A STOCHASTIC MODEL FOR JOINT RECEPTION, STAGING, ONWARD MOVEMENT, AND INTEGRATION (JRSOI)

THESIS

Nathan P. Sherman, Captain, USAF

AFIT/GOR/ENS/03-21

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense or the United States Government.

AFIT/GOR/ENS/03-21

A STOCHASTIC MODEL FOR JOINT RECEPTION, STAGING, ONWARD MOVEMENT, AND INTEGRATION (JRSOI)

THESIS

Presented to the Faculty
Department of Operational Sciences
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Operations Research

Nathan P. Sherman, B.S.
Captain, USAF

March 2003

AFIT/GOR/ENS/03-21

A STOCHASTIC MODEL FOR JOINT RECEPTION, STAGING, ONWARD MOVEMENT, AND INTEGRATION (JRSOI)

Nathan P. Sherman, B.S.

Captain, USAF

Approved:

Dr. Jeffrey P. Kharoufeh Thesis Advisor	Date
Dr. James T. Moore Reader	Date

Acknowledgments

The first-rate insight, guidance, and editing skills as well as encouragement from my advisor, Dr. Jeff Kharoufeh, contributed greatly to the quality and reach of this research. Similarly, the careful editing and experienced direction on joint mobility modeling issues provided by my reader, Dr. Jim Moore, had a significant and positive impact.

The United States Transportation Command (USTRANSCOM), through the office of Lieutenant Colonel Robert Brigantic, Operations Analyst, provided guidance, contacts, and travel funds to support this effort. The insight provided by the following contractors at USTRANSCOM was very helpful: Dr. Jean Mahan, Jay Marcotte, and Gary Masters. A principle contact, Lieutenant Colonel David Kolleda, Director of the Deployment Process Modernization Office, Fort Eustis VA, provided several very informative briefings on JRSOI.

Major John Gage, a faculty member of the AFIT's Logistics School of Management, planted the seed for this research and shared a great deal of first-hand knowledge of the JRSOI process. He provided multiple contacts for people possessing first-hand knowledge and experience of JRSOI. The insights and information links provided by Lieutenant Colonel Steven Lawlor, former commander of the 51st Logistics Readiness Squadron in the Republic of Korea, were particularly useful.

I am grateful to God for paving the path before me and sustaining me and to my wife for her patience and support during the arduous research process. Finally, I acknowledge Jesus Christ, the King of Kings and Lord of Lords, whose name is the only given under heaven whereby man might be saved.

Nathan P. Sherman
March 2003

Table of Contents

Page

List of Figures

List of Tables

Table Page

AFIT/GOR/ENS/03-21

Abstract

A stochastic model for the performance evaluation of a key phase in the deployment process, namely Joint Reception, Staging, Onward Movement, and Integration (JRSOI) is presented. The process is modeled as an open, multi-class tandem queueing network wherein personnel and various classes of cargo are modeled as the flow entities and the stages of the process constitute individual queueing stations. Single- and multiple-class models at both low and high resolutions are presented. No analytical, stochastic model of this process currently exists in the literature or in practice. The model provides a quick look at key aggregate performance measures such as system throughput and closure, and can be used to expediently identify problems occurring during JRSOI and the impact they have on the process. This information can substantially aid decision makers in regulating process flow. The queueing network model developed here can easily be expanded and adapted to any potential area of conflict. Numerical comparisons with Monte-Carlo simulation demonstrate that the model provides a viable, novel approach to the problem.

A STOCHASTIC MODEL FOR JOINT RECEPTION, STAGING, ONWARD MOVEMENT, AND INTEGRATION (JRSOI)

1. Introduction

When the U.S. military responds to a contingency, it must assemble the necessary personnel, equipment, and materiel and move them to the area of operations (AO). This process is commonly referred to as deployment. The U.S. military conducts deployment in four phases: preparation activities, movement to and activities at ports of embarkation (POEs), movement to ports of debarkation (PODs), and movement from PODs to (and activities at) tactical assembly areas (TAAs). The latter phase is the main focus of this research. This phase is referred to as Reception, Staging, Onward Movement, and Integration (RSOI). Because deployment typically involves a joint effort of the military services, this phase is often referred to as *Joint* Reception, Staging, Onward Movement, and Integration (JRSOI) [18]. (The terms JRSO&I, RSO&I, JRSOI, and RSOI are used interchangeably in this thesis.) This thesis explores analytical, stochastic models for the performance evaluation of JRSOI.

1.1 Problem Background

JRSOI is the essential process that transitions deploying personnel (pax) and equipment and materiel (cargo) into combat-ready forces in the AO. It includes receiving pax and cargo at PODs, assembling them into units at designated staging sites, moving these units to destinations within the area of conflict, and integrating them into combat-ready joint fighting forces. One primary goal of JRSOI is to achieve flow balance, meaning the flow of pax and cargo is directed at a rate that can be accommodated at every stage of the process from arrival at a POD to final integration within a combat unit [9]. However, this goal is seldom realized in military exercises and real world conflicts [8]. Saturation often occurs at various points in the process because flow is not being directed at a sufficient rate or due to resource constraints [9]. Saturation strains available processing capacity and provides the enemy a vulnerable target over an extended period. In what follows, some relevant background is given for modes of transport into PODs, POD classification, and types of entry into an area of conflict.

There are two modes of initial transport for pax and cargo being deployed: airlift and sealift. Assets delivered by airlift are received at Aerial Ports of Debarkation (APODs) and those delivered by sealift are received at Seaports of Debarkation (SPODs). Airlift is expedient, requiring hours or days, and best suited for transport of light, early-entry forces and for pax in general. Sealift is the most economical means of moving bulk goods and heavy equipment but is extremely slow, requiring weeks or even months. Typically, all pax and cargo arrive by airlift during the first three weeks of deployment [8]. After this initial period, pax and high priority cargo generally arrive by air and bulk cargo by sea [17]. Historically, for both major and minor contingencies, 90 percent of all cargo by weight has been transported by sea, with the remaining 10 percent transported by air [8]. This thesis takes into account pax and cargo arriving by both airlift and sealift.

There are three types of SPODs: improved, world class ports; unimproved or degraded ports; and bare beach or no port environment [8]. APODs are not classified by type but have a similar wide variance in modernization and capability. This thesis models a well-established and modernized APOD as the physical location for the reception stage while also taking into account assets arriving through neighboring APODs and SPODs. There are two types of entry into an area of conflict: opposed and unopposed [8]. For opposed entry, combat units arrive intact, ready to fight immediately upon arrival. Unopposed entry involves transporting pax and cargo to

APODs and SPODs in non-hostile territory and moving them through the JRSOI process. This thesis focuses on unopposed entry.

Successful deployment has traditionally been measured by the ability to move forces from ports of embarkation to ports of debarkation [8]. As a result, pax and cargo have arrived at PODs in sporadic surges leading to bottlenecks and other inefficiencies during the JRSOI process [8]. In this thesis, JRSOI is modeled to provide insight and analysis that aid in the planning and execution of future U.S. military deployments.

1.2 Problem Definition and Methodology

The current JRSOI process experiences serious bottlenecks during military operations and exercises due to insufficiently regulated flow between its stages [24]. This thesis seeks to create an analytical, stochastic model of the process and to formally analyze this model for the purpose of obtaining computationally expedient estimates of the appropriate performance measures. To the author's knowledge, there does not exist a formal, analytical model that captures the aggregate features of the JRSOI process. Next, the research objectives of this work are described in greater detail.

The first objective of this research is to gain a thorough understanding of the JRSOI process by speaking with the subject matter experts (SMEs) associated with JRSOI and studying relevant Department of Defense (DoD) publications. This facilitates establishment of a concrete notional JRSOI process on which to base an analytical model. The next objective is to provide a determination of the most important performance metrics for the overall process and develop a stochastic model using a queueing network approach. The third and final objective is to analyze the model and conduct validation and verification testing. In what follows, the methodologies to be employed in this thesis are discussed.

The notional JRSOI process shall be modeled as an open queueing network. The network consists of four nodes in tandem representing the four main stages of the process: reception, staging, onward movement, and integration. These four nodes are subsequently decomposed into a subnetwork encompassing the aggregate features of each stage. Queueing network analysis software shall be developed to obtain performance measures. The software is designed to allow any parametric probability distribution to be used for interarrival and service times at each queue. Hence, for any $G/G/m$ queues in the network, performance measures are approximate rather than exact. Comparison of the results obtained using queueing network analysis with computer simulation results demonstrate the accuracy of these approximations. Next, the contributions of this work are discussed.

This thesis contributes significantly to the military operations research literature. It provides a novel approach for modeling JRSOI. No analytical, stochastic model of this process currently exists in the literature or in practice. The fact that the model developed in this thesis can be coded using readily available software languages such as Visual Basic and MATLAB means that no costly simulation or optimization software is required to run it. The approximation equations used by the model enable the computation of precise estimates for key performance metrics, such as throughput and mean time in system, in a fraction of the time it would take a simulation. The model can be used to expediently identify when and where problems occur during JRSOI and to what degree these problems affect the process. This information can substantially aid decision makers in regulating process flow. The base model developed here can readily be expanded and adapted to any potential area of conflict. In addition, this thesis contributes to the stochastic operations research community. In general, it demonstrates the importance of capturing stochasticity in real-world systems. Furthermore, it provides a novel application of well-known techniques. In particular, it validates the use of approximation techniques for obtaining performance metrics as opposed to obtaining exact measures for less exact network-models based on simplifying assumptions. In summary, these techniques provide more accurate results and are much easier to apply.

1.3 Thesis Outline

The next chapter contains a broad review of available literature and knowledge from SMEs. First, the JRSOI process is investigated to provide the understanding needed to build a model. Second, queueing network analysis techniques that allow for efficient computation of highly accurate results for queuing network models are investigated.

Chapter 3 defines the problem and gives a rationale for the approach taken to solve it. It contains an explicit development of the queueing network approximation equations required for development of the queueing network models. Four queueing network model configurations are developed based on four different class (single versus multiple) and resolution (high versus low) combinations.

Chapter 4 contains a numerical analysis of the models developed in Chapter 3. It also contains the development of computer simulation models for the same networks. The results from the analytical and simulation models are statistically compared. Finally, Chapter 5 gives concluding remarks including insights gained from the conducted tests and analyses, summary of contributions to the academic and military communities, and recommendations for future research.

2. Review of the Literature

In this chapter, a review of pertinent literature is provided. First, a brief look at the history of JRSOI is provided to establish an appropriate context. This covers the origins of JRSOI as well as doctrinal and tool development.

2.1 JRSOI Background

JRSOI was developed to doctrinally combine, into a single, formalized process, the tasks of the receiving and organizing of incoming troops and equipment at PODs, moving them to TAAs, and integrating them with existing combat units [8]. "JRSOI [was] designed to eliminate much of the confusion associated with people and cargo arriving in theater in disorganized pieces and to break down the bottlenecks that have historically existed in large-scale joint operations" [27].

Deployment success has traditionally been measured by the rate at which pax and cargo are delivered to PODs, and as a result, the tasks encompassed by JRSOI have generally been overlooked [8]. The evolution of military doctrine, combined with lessons learned from the Gulf War, led to a focus on and formalization of JRSOI starting in the early 1990s [14]. In terms of doctrine, the post-cold war era has seen a transition from forward basing to the "force projection" of expeditionary forces. This translates to a shift between having assets based near potential AOs to having them delivered as needed to AOs by air and sea through PODs. The term *force closure* has been redefined as the point at which assets are successfully integrated into combat units at TAAs rather than the point at which they arrive at PODs. In terms of lessons learned from the Gulf War, Lieutenant General (Ret.) Joseph M. Heiser summarized by stating, "As we have seen many times, the U.S. can ship supplies and material to an objective area much more effectively and efficiently than the objective area can unload and distribute those supplies" [9]. One of the main causes of this deficiency was that "deployment planning tools in 1990-91 did not address distribution within the theater" [27]. Another primary reason was the poor in-transit visibility of assets [7]. These problems combined to make achieving unit integrity very difficult [21]. They also contributed significantly to the build up of pax and cargo at PODs and staging areas which stressed the Saudi infrastructure and exposed troop concentrations to possible enemy attack [9]. To alleviate these problems, in-theater flow of pax and cargo must be planned, controlled, and tracked.

As part of the formalization of JRSOI, large-scale exercises have been developed and conducted. The National Training Center (NTC), Fort Irwin, California, began to develop an RSOI training program in 1994 and has subsequently integrated RSOI tasks into its exercises at Fort Irwin and elsewhere [15]. The first joint military *RSOI Exercise* was held in South Korea in April 1994 [25] and continues to be held annually. *Foal Eagle*, another U.S.-South Korean joint military exercise, held annually since 1961, began to focus on RSOI aspects in 1995 [6] and was linked with the *RSOI Exercise* in 2002. An RSOI training center was established in Germany during Bosnia operations in the mid 1990s [26]. In 1996, the *U.S. Army Force Deployment Rock Drill Exercise* was conducted at Fort Eustis, Virginia, by the Army to increase understanding of RSOI and to aid its development [22]. "Since the Gulf War, RSOI has become one of the most important missions for U.S. Army Forces Central Command–Kuwait (ARCENT-KU)" [16]. This mission for perfecting the RSOI process [1] was formalized in 1999 when ARCENT-KU first conducted *Lucky Sentinel*, a joint annual deployment exercise involving the U.S., U.K., and Kuwait [23]. ARCENT–Qatar has a similar, RSOI-focused mission [2]. The efforts and activities mentioned here highlight, but do not exhaust, the efforts and activities related to JRSOI.

RSOI was derived from U.S. Army Field Manual 100-17 (28 Oct 92), Mobilization, Deployment, Redeployment, and Demobilization [3]. "A series of Joint and Army workshops in 1995 accelerated doctrine development" [15]. The U.S. Army Field Manual 100-17-3, Reception, Staging, Onward Movement, and Integration was first published as a working draft in 1995 and formally published in 1999. The U.S. Army Transportation School, Fort Eustis, Virginia, has had “the Training and Doctrine Command lead for development of RSO&I doctrine” [15]. Experience gained from small scale contingencies in the 1990s “indicates that the same lessons were relearned during each operation because the responsibilities for improving the deployment process were diffused among many different organizations and not focused on the requirements of the joint force” [5]. Therefore, to tie deployment improvement efforts together, the Deployment Process Special Action Group (DPSAG) was established in 1996 and “institutionalized as the Deployment Division in the Directorate for Logistics (J4) of the Joint Staff in 1997” [5]. In 1998, the Commander-in-Chief for the United States Joint Forces Command (CINCUSJFCOM) was designated “the joint deployment process owner for DoD” [5]. As a direct result, the first joint doctrine on RSOI, Joint Publication 4-01.8 Joint Tactics, Techniques, and Procedures for Joint Reception, Staging, Onward Movement, and Integration, was published in 2000 [5]. The Joint Deployment Training Center (JDTC), as an element of USTRANSCOM, is the DoD’s center for learning and information for all joint deployment doctrine, education and training issues [18].

2.2 Modeling and Simulation Tools

Doctrinal development has been complemented by the development and refinement of a host of modeling and simulation (M&S) tools to enhance force projection planning and execution. One of the most prevalent modeling tools for the JRSOI portion of deployment is the Enhanced Logistics Intra-theater Support Tool (ELIST). It is designed to evaluate the feasibility of surface transportation movement plans, to simulate the marry-up of unit equipment and troops in staging areas, and to estimate closure trends [14]. Other tools that have been developed to model JRSOI, or various aspects of it, include TRANSCAP, SMARTBRIDGE, MATT, CITM, and FASTALS [14]. However, these tools address limited portions of JRSOI and have only limited capability to interact and share data [22]. According to the Warfighter M&S Assessment, conducted by U.S. Central Command, U.S. Southern Command, and U.S. Special Operations Command in August 2000, "...modeling of the Reception and Staging Onward-Movement Integration (RSOI) process, from method of delivery through theater to tactical assembly area, is not done well." This inability to efficiently model JRSOI functions hinders the ability of combatant commanders to determine feasibility of operational plans [24]. Thus, despite recent M&S development efforts, much work is yet required to establish tools that can efficiently and effectively aid in the planning and execution of military operations [14].

Furthermore, for JRSOI-specific tools to aid in overall deployment planning and execution, they must be integrated with tools modeling the other phases of deployment. Several full-spectrum M&S tools have been, and continue to be, developed to enable planners and operators to address end-to-end mobility requirements covering the whole spectrum of deployment [22]. Some of these are standalone while others are suites that tie together a collection of existing tools. The most prevalent of these is the Analysis of Mobility Platform (AMP) suite which incorporates a host of prevalent tools including ELIST [14]. However, none of the end-to-end tools developed thus far has the capability to authoritatively provide comprehensive end-to-end analysis [14]. As a result, the Department of Defense continues to experience difficulty in deployment planning and execution capability [14]. Further explanation and description of current deployment M&S tools, their interaction, and ongoing development efforts is given in a survey by Drummer [14].

2.3 Data Management Systems

One of the key capabilities of deployment M&S tools is the ability to interact with and manage databases containing planning requirement and asset tracking information. In particular, successful deployment depends on a well-planned and carefully managed Time-Phased Force Deployment Data (TPFDD) flow [9]. The

TPFDD is a planning database that indicates force requirements and prioritizes transportation movement [8] produced by the Joint Operation Planning and Execution System (JOPES) [21]. Its purpose is to phase forces into operational areas at the times and places required to support the plan of operations [10]. It is based on the supported CINCs statement of his/her requirements by unit type, time period, and priority for arrival [11]. The TPFDD must be analyzed to determine feasibility and must be updated to incorporate any changes. The main JRSOI tool that supports TPFDD feasibility analysis and has the ability to update it is ELIST. However, as it and similar systems have become increasingly more capable, they have also become increasingly more complex. This has lead to extensively long run times and complicated operating procedures. The developers of ELIST are working to make their "model more user-friendly so it can operate without extensive contractor support" [24]. As with the M&S tools, the TPFDD and other deployment databases suffer because a coordinated effort to link them together is lacking [24].

Databases containing asset visibility and tracking information are fed by Automated Identification Technology (AIT) tools. "Commands operating PODs have historically been unaware of some inbound equipment or cargo. The lack of asset visibility slows the reception process and chokes port capabilities" [17]. During the Gulf War, the "inability to keep track of inbound shipments [was] evident in most of the [Logistics Reports]" [27]. "Without in-transit visibility, logisticians could only track, not predict, the logistics situation. Asset visibility was also a problem: of the more than forty thousand containers deployed to the theater, well over half had to be opened at least once to determine contents, ownership, and destination" [27]. Accurate In Transit Visibility (ITV) will "provide decision makers an increased capability to react to unexpected bottlenecks and verify unit movements" [21]. It can also give inprocessing "teams a much better opportunity to know what will arrive, when it will arrive, and prepare timely plans for unloading, marshalling, and" other activities as pax and cargo arrive at PODs [21]. AIT can provide commanders information on the status, location, and movement of assets. It can include "bar codes for individual items, optical memory cards for packs and containers, radio frequency tags for containers and pallets, and movement tracking capability that uses satellite links for convoys, trains, and barges" [12]. "This information can enhance theater RSO&I management capabilities and enable efficient reception and processing of units and equipment as they flow through the RSO&I process" [17]. In particular, AIT can capture arrival and departure data for pax and cargo at PODs, moving through Intermediate Staging Bases (ISBs), and at other designated transit points [17]. It can employ satellite tracking during onward movement and capture data on the arrival and force closure of units and equipment at TAAs [17]. The AIT tools used for deployment have been incorporated into the JOPES.

The JOPES is the integrated command and control system used by the U.S. military to plan, execute, and monitor joint military operations [12]. It "provides users with an ordered and comprehensive set of policies and procedures for solving complex mobility force deployment and sustainment problems" [10]. It has an information reporting structure supported by automated data processing on the Global Command and Control System (GCCS) [10]. GCSS is designed to provide accurate and near real time total asset visibility vital to the deployment, employment, sustainment, reconstitution, and redeployment of joint combat assets or resources [11]. It is a key tool for commanders in planning and conducting joint operations [12]. The principle enablers of GCCS are the Global Transportation Network (GTN) and Joint Total Asset Availability (JTAV) systems. The GTN is the designated Department of Defense (DOD) in-transit visibility (ITV) system, providing customers with the ability to track the identity, status, and location of DOD units and non-unit cargo, passengers, patients, forces, and military and commercial airlift, sealift, and surface assets from origin to destination across the range of military operations [10]. In the future, GTN is expected to provide advanced feasibility analysis of planned surface and air movements [21]. JTAV is the DOD's automated capability for timely and accurate information on the location, movement, status, and identity of units, personnel, equipment, and supplies [18]. JTAV integrates in process, in storage, and In Transit Visibility (ITV). The development efforts for the systems mentioned in this paragraph were initiated before the Gulf War but did not gain momentum until after the War.

2.4 Summary

The recent emergence of JRSOI reflects the increased focus on deployment in general, and its final phase in particular, integral to new force projection doctrine. Many modeling and simulation and data management tools have been developed to aid in the planning and execution of deployments. Effective modeling and simulation tools must exploit the data management tools. Ideally, some of these tools should have the capability to support the "rapid, accurate projection of logistic capabilities and bottlenecks" [21]. Currently, however, tools having the capability to manage and predict the flow of deployed assets, are either slow and cumbersome or nonexistent.

These functions can be performed more efficiently by analytical tools that incorporate the stochastic nature of the operations. However, to the author's knowledge, no such tools exist for JRSOI or the deployment process in general. This research seeks to build a foundation for such an analytical model. A tool based on this model can provide key decision makers tremendous insight into how to manage asset flow during JRSOI with comparatively little computational effort. In Chapter 3, the formal stochastic model for JRSOI is presented. The selected approach is based on the well-known theory of queues and queueing networks.

3. Methodology

In this chapter, an analytical, stochastic model for the Joint Reception, Staging, Onward Movement, and Integration (JRSOI) process is proposed. This is followed by a review of some well known results for queueing network models.

3.1 Description of the Approach

The JRSOI process often experiences bottlenecks due to insufficiently regulated flow. A primary reason for this insufficient regulation is that information for how pax and cargo flow through the process is lacking. This lack of such information is caused by the absence of analytical tools to approximate it. This research provides an analytical method for calculating these critical performance measures.

A primary goal of JRSOI is flow balance, the continuous and controlled flow of pax and cargo within and between its stages. Flow at a particular stage is based on the interaction between arriving pax and cargo and the service capabilities of that stage. Since interarrival and service times have a degree of variability that can significantly impact flow, their behavior needs to be captured using probability distributions. This inherent randomness of the process lends itself to stochastic modeling. Furthermore, since the process can be easily decomposed into sequential stages, and interarrival and service times can be observed for each of these stages, it is highly appropriate to model this process as a tandem queueing network.

The JRSOI queueing network model was developed at two levels of resolution. The lower resolution configurations consist of four nodes, one for each stage of the process. The higher resolution configurations replace each of the four nodes of the lower resolution model with a subnetwork of nodes representing the process within that node's respective stage. The entities for the model are pax and cargo. In addition to having two levels of resolution, the model was developed at two entity classification settings: single-class and multi-class. For the single-class configurations, all entities are treated equivalently. For the multi-class configurations, entities are separated into four classes: pax, outsize cargo, oversize cargo, and bulk cargo. The cargo classifications are based on how the cargo is transported and on what aircraft it may fit. Bulk cargo is assembled and transported on a single pallet; pallets can fit on any aircraft. Oversize cargo is too large to fit on single pallet but can fit on any aircraft. Some pieces of oversize cargo can be assembled onto a train of pallets for transport. The final class is outsize; because of its size and dimensions, it can only be carried on a C-17 or C-5. In total, four model configurations were developed and tested, one for each resolution and entity-class combination. Next, the individual stages of the process are discussed.

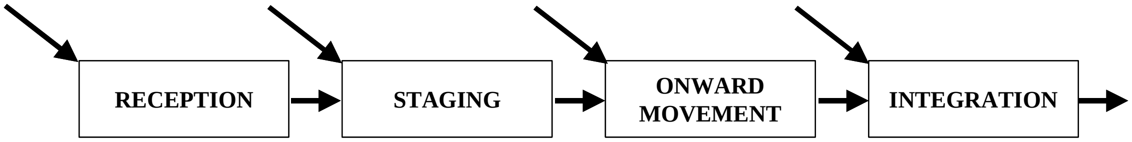

Figure 3.1 A graphical depiction of the JRSOI process.

Reception is the process of receiving, offloading, and marshalling pax and cargo at a port of debarkation and then transporting them to a staging area. The network model's reception node represents a single Aerial Port of Debarkation (APOD). Entities for this node arrive in bulk on airplanes. Upon arrival, these packets of entities form a queue awaiting service at the APOD. The server sequentially receives, offloads, marshals, and transports these entities to the staging area where they await transport by bus, train, or other mode of transportation. (Note that the model allows entities to enter the network at any node thus accounting for the merging in of pax and cargo from neighboring APODs and Sea Ports of Debarkation (SPODs).)

Staging is the process of assembling pax and cargo into mission ready units and preparing them for onward movement. The staging node represents an Intermediate Staging Base (ISB). Entities arrive in bulk from the reception node and from other APODs and SPODs. Upon arrival, these packets of entities form a queue awaiting service. The server assembles these entities into mission-ready units, prepares them for onward movement, and places them in a holding area to await transportation.

Onward movement is the process of moving units and accompanying materiel from the staging area to Tactical Assembly Areas (TAA) or other final destinations in the AO. The onward movement node represents various modes and routes of transportation. The queue is the holding area at the staging node. The model allows entities to enter this holding area from external sources. The server moves entities to the TAAs by rail, road (typically in convoy), inland or coastal waterway, or air.

Integration is the process of transferring mission-ready units into the combatant commander's force. The integration node represents the TAAs, taken collectively. Entities arrive in bulk from the onward movement node and external sources. Upon arrival, these packets of entities arrive and form a queue awaiting service. The server transfers mission-ready units into the combatant commander's force. Once units complete processing at this stage, the entities are assumed to exit the system. Next, modeling techniques for stochastic queueing networks are discussed.

Stochastic networks are typically modeled using computer simulation or analytical techniques. Computer simulations for networks can be relatively easy to develop and test if commercial simulation software is used. For some networks, these simulations can expediently produce highly accurate results given the high speed of modern computers. However, for complex networks, they can take hours or days to run if a high level of precision is required. In addition, commercial simulation software can be expensive (sometimes prohibitively so) for the end user to purchase. As an example, a single, one-year license for the simulation software used for this research has an approximate purchase price of $16K. On a much larger scale, the U.S. military often hires contractors to develop sophisticated simulation models such as those reviewed in the previous chapter. These efforts cost the military millions of dollars and the models developed are often so complex that they require extensive contractor support to operate [14]. Run times for these models are measured in hours or days. Alternatively, a well developed tool based on analytical queueing techniques can be just as easy to use and produces results much faster, typically in a few seconds or less. Another advantage of such techniques is that they are simplistic enough to be programmed in languages commonly found on personal computers such as Visual Basic® or MATLAB®. In this thesis, a model is built based on analytical queueing techniques for the purpose of obtaining accurate network performance metrics in an expedient manner and at negligible cost as compared to simulation. Computer simulations are developed as a benchmark for comparison purposes.

There are two primary approaches for obtaining performance measures of a network when using analytical queueing techniques. One approach is to approximate non-exponential interarrival- and service-time distributions with exponential distributions or sums of exponential distributions. By this approach, exact analytical results can be obtained for the model, but the validity of the model often suffers. The other approach is to approximate performance measures based on characteristics, such as moments of probability distributions. By this approach, approximate results can be obtained for highly valid models. The goal then is to ensure the approximating equations produce accurate results. The latter approach is the approach taken here. Furthermore, the equations used to estimate performance measures are taken from Whitt's Queueing Network Analyzer (QNA) development [28]. Whitt's QNA equations were chosen because of their simplicity and high level of demonstrated accuracy. Next, the formal description of the JRSOI model is provided.

3.2 Formal Model Description

In this section, the formal mathematical model is developed. First, a rudimentary review of queueing theory is provided. For more details, the reader is referred to Kleinrock [20]. Next, foundational assumptions and definitions are established for the queueing network model. Finally, additional assumptions and characteristics unique to each of the four specific model configurations are given.

A queueing system may be comprised of a single queueing station or a network of such stations. Therefore, to understand a queueing network, the characteristics of the single queueing station(s) comprising that system must be described. There are five components to a queueing station: an arrival process, a service process, the number of servers, the system capacity, and a queueing discipline. The arrival process is primarily characterized by a parameterized probability distribution describing the interarrival times of entities. Entities may be of a single class or of multiple classes, and may arrive one at a time or in batches. Additional attributes that sometimes impact the characteristics of an arrival process, such as *balking*, *jockeying*, and *reneging*, are not discussed in this thesis. The service process is characterized by a parameterized probability distribution describing the service times of entities. Service rates may or may not depend upon factors such as entity class or the number of entities in the system. The third component of a queueing station, the number of servers, simply indicates how many servers, operating in parallel, are available to service entities. The fourth component, system capacity, is the total number of entities that can be accommodated by the system. It is the sum of the number of entities that can be accommodated by the server(s) and by the queue. It may be finite or infinite. The final component, queueing discipline, determines the order in which entities are served. Common disciplines include first come, first served (FCFS), last come, first served (LCFS), non-preemptive priority (NPP), preemptive resume (PR), and service in random order (SIRO). Queueing disciplines that determine service order based on entity classification are called *priority* queueing disciplines. Queueing stations are succinctly described using Kendall's notation [19], $A/B/C/D/E$, where A specifies the interarrival-time distribution, B, the service-time distribution, C, the number of servers, D, the system capacity, and E, the queueing discipline. The terms D and E need not be specified in which case D would default to ∞ (signifying infinite capacity) and E would default to the FCFS queueing discipline. Based on this information, performance measures for a queueing station can found.

For performance measures to be derived, a systematic method must be developed to capture, over time, the number of entities at a queueing station. Let the integer-valued random variable $X(t)$ be the number of entities at a queueing station at time t. Then the sequence $\{X(t) : t \geq 0\}$ is a continuous-time stochastic process on state space $S = \{0, 1, 2, ...\}$ that captures the number of entities at the station over time. The performance measures of interest in this thesis are based on the long-run or 'steady-state' performance of the system. Hence, mean performance values are sought as $t \longrightarrow \infty$. Define $p_z = \lim_{t\to\infty} P\{X(t) = z\}$ as the steady-state probability that there are z entities in the system. Define ρ to be the *traffic intensity* or *utilization* which indicates the steady-state proportion of time that the servers are not idle. It can be shown the probability that the servers are idle, $(1 - \rho)$, is equivalent to the probability that the system is empty, p_0. The steady-state distribution of $X(t)$ is given by the p_z defined earlier where the p_z are usually calculated recursively based on p_0 and the parameters of the interarrival- and service-time distributions. The primary performance measures of interest can be derived based on the p_z. These measures are W_q, W, L_q, and L. Their form can vary widely depending on the specifications of the queueing station. Define $L = \lim_{t\to\infty} E[X(t)]$ as the mean number of entities at the queueing station. Then $L_q = L - \rho$ is the mean number of entities in the queue. The mean time an entity spends at the queueing station, $W = \lambda L$, and the mean time an entity spends waiting in the queue, $W_q = \lambda L_q$, can be found using *Little's Law* where λ is the arrival rate. In general, once any of these four measures is found (based on the p_z), the remaining three can be found directly using Little's Law and the parameters of the interarrival- and service-time distributions, λ and μ, respectively. In this thesis, W_q is found first and W, L_q, and L are calculated from W_q using Little's Law and the process parameters. Little's Law holds for all *work-conserving* queueing disciplines which are those that never allow servers to sit idle when there is work to be done.

A queueing network is a collection of two or more queueing stations in which customers flow between the stations. It can be represented by a connected graph in which each node represents a queueing station and each arc represents an entity flow having nonzero probability.

Next, the foundational assumptions and definitions for the model are established. The network is open, meaning entities are allowed to enter and depart from the network. There are no capacity constraints on the number of entities in the system and each node has unlimited waiting space. Every node is modeled as a $G/G/m$ queue. The first G indicates that the interarrival times are generally distributed, the second G indicates that the service times are generally distributed, and the m indicates the finite number of servers at each queueing station. (An M (*Markovian*) designation for interarrival- and service-time distributions indicates use of the exponential distribution.) Allowing interarrival and service times to be generally distributed is one of the key features of the model. This allows the specification of any probability distribution for modeling arrival and service processes. Thus, the model does not restrict usage to the exponential distribution. Furthermore, this model does not require explicit characterization of the probability distribution, but only the mean and variance. The model allows both internal and external arrivals to each node (as depicted in Figure 3.1). The interarrival-time distribution parameters calculated for each node incorporate both internal and external arrivals.

QNA is applied to approximate the desired performance metrics. The general methodology is parametric decomposition or product-form-solution where each node is analyzed separately based on the internal flow parameters calculated. The internal flow parameters attempt to capture any dependence that exists among the nodes. The decomposition approximation of Whitt [28] provides more accurate results for non-Markovian networks than would be acquired by calculating exact results for Markovian representations of these networks. It is important to note that the approximation equations used here give exact analytical results for Markovian, single-class networks.

As an approximation, the arrival processes are treated as renewal processes. A *renewal* process is a sequence of independently and identically distributed nonnegative random variables $\{X_n\,,\, n \geq 1\}$ such that X_n denotes the time between the $(n-1)$st and nth arrivals and the random variable $T_n = X_1 + X_2 + \cdots + X_n$ denotes the time of the nth arrival. The significance of an arrival process being a renewal process is that it exhibits Markovian behavior without being exponentially distributed and enables characterization using familiar parameters such as mean and squared coefficient of variation. For this thesis, these two particular parameters of the arrival processes are approximated by the equivalent parameters of their associated renewal processes.

To approximate the traffic rates and standard performance measures for the network, two parameters are required for both the arrival process and the service process of each node. The two arrival process parameters required are the arrival rate λ_j and squared coefficient of variation c_{aj}^2 for node j. The two service process parameters required are the service rate μ_j and squared coefficient of variation c_{sj}^2 for node j. In addition, a routing matrix $R = [r_{i,j}]$ is required giving the proportion of entities completing service at facility i that go next to facility j. An external node, node 0, is defined to represent the world external to the network.

The two service process parameters are given directly as input. However, the interarrival-time parameters are calculated based on the external arrival rates $\lambda_{0,j}$ and the routing matrix R using the well-known traffic rate equations. In total, four numbers for each node in the network are required as input to run the model. Note that all these equations can ultimately be written in terms of the input parameters.

Suppose the network has n nodes. Then the fundamental traffic-rate equations are

$$\lambda_j = \lambda_{0,j} + \sum_{i=1}^{n} \lambda_i r_{i,j}, \qquad j = 1, 2, ..., n \tag{3.1}$$

or in matrix notation,

$$\Lambda = \Lambda_0 (I - R)^{-1} \tag{3.2}$$

where $\Lambda_0 = [\lambda_{0,j}]$ is the external arrival-rate vector and R is the routing matrix. Using these arrival rates, the traffic intensities (the probability that the server is busy at an arbitrary point in time) defined by

$$\rho_j = \lambda_j / \left(m_j \mu_j\right), \qquad j = 1, 2, ..., n \tag{3.3}$$

can be solved where m_j indicates the number of servers at node j. The stability condition, $\rho_j < 1$, is required for each node j. The related quantities for the arcs can now be computed where

$$\lambda_{i,j} = \lambda_i r_{i,j} \tag{3.4}$$

is the arrival rate to node j from node i and

$$p_{i,j} = \lambda_{i,j} / \lambda_j \tag{3.5}$$

is the proportion of arrivals to node j from node i. Similarly, the departure rate out of the network from node i can be calculated:

$$d_i = \lambda_i \left(1 - \sum_{j=1}^{n} r_{i,j}\right). \tag{3.6}$$

The traffic variability equations are found by solving the system of equations given by

$$c_{aj}^2 = a_j + \sum_{i=1}^{n} c_{ai}^2 b_{i,j}, \qquad j = 1, 2, ..., n. \tag{3.7}$$

where a_j and $b_{i,j}$ are constants, depending on the input data:

$$a_j = 1 + w_j \left\{ \left(p_{0,j} c_{0j}^2 - 1\right) + \sum_{i=1}^{n} p_{i,j} \left[(1 - r_{i,j}) + (1 - v_{i,j}) \, r_{i,j} \rho_i^2 x_i\right] \right\} \tag{3.8}$$

and

$$b_{i,j} = w_j p_{i,j} r_{i,j} \left[v_{i,j} + (1 - v_{i,j}) \left(1 - \rho_i^2\right)\right], \tag{3.9}$$

where x_i and w_j are defined as

$$x_i = 1 + m_i^{-0.5} \left(\max\left\{c_{si}^2, 0.2\right\} - 1\right), \tag{3.10}$$

and

$$w_j = \left[1 + n \left(1 - \rho_j\right)^2 (v_j - 1)\right]^{-1}, \tag{3.11}$$

with

$$v_j = \left(\sum_{i=0}^{n} p_{i,j}^2\right)^{-1}. \tag{3.12}$$

The v_{ij} values are adjustable values used in the convex combinations above to approximate departure operations. They are set to zero for this research to test the performance of QNA without the assistance of this fine-tuning feature.

Now that the traffic intensity ρ_j and the rate λ_j and variability c_{aj}^2 parameters for the internal arrival processes have been found for $j = 1, 2, ..., n$, the performance measures for each node can be calculated. These aggregate network performance measures are obtained based on the approximating assumption that the network nodes are stochastically independent given the approximate flow parameters. Recall that the rate μ_j and variability c_{sj}^2 parameters for the service process at each node j were given as input.

The steady-state mean waiting time in queue, W_{q_j}, for node j is

$$W_{q_j} = \frac{\rho_j \left(c_{aj}^2 + c_{sj}^2\right) g_j}{2\mu_j \left(1 - \rho_j\right)}, \qquad j = 1, 2, ..., n. \tag{3.13}$$

where $g_j \equiv g_j\left(\rho_j, c_{aj}^2, c_{sj}^2\right)$ is defined by [28]

$$g_j\left(\rho_j, c_{aj}^2, c_{sj}^2\right) = \begin{cases} \exp\left[\frac{-2(1-\rho_j)}{3\rho_j}\frac{\left(1-c_{aj}^2\right)^2}{c_{aj}^2+c_{sj}^2}\right] & c_{aj}^2 < 1 \\ \\ 1 & c_{aj}^2 \geq 1. \end{cases} \tag{3.14}$$

By Little's Law, the steady-state waiting time W_j, number of entities in each queue L_{q_j}, and number of entities at each node L_j can be calculated:

$$W_j = 1/\mu_j + W_{q_j}, \tag{3.15}$$

$$L_{q_j} = \lambda_j W_{q_j}, \tag{3.16}$$

$$L_j = \rho_j + \lambda_j W_{q_j}. \tag{3.17}$$

The probability of delay (the probability a customer must wait in a queue for service) at each node is denoted by σ_j which Whitt found by using the Kraemer and Langenbach-Belz approximation [28]:

$$\sigma_j = \rho_j + \left(c_{aj}^2 - 1\right)\rho_j\left(1-\rho_j\right)h_j, \tag{3.18}$$

where $h_j \equiv h_j(\rho_j, c_{aj}^2, c_{sj}^2)$ is defined by [28]

$$h_j\left(\rho_j, c_{aj}^2, c_{sj}^2\right) = \begin{cases} \frac{1+c_{aj}^2+\rho_j c_{sj}^2}{1+\rho_j\left(c_{sj}^2-1\right)+\rho_j^2\left(4c_{aj}^2+c_{sj}^2\right)} & c_{aj}^2 < 1 \\ \\ \frac{4\rho_j}{c_{aj}^2+\rho_j^2\left(4c_{aj}^2+c_{sj}^2\right)} & c_{aj}^2 \geq 1. \end{cases} \tag{3.19}$$

Finally, the performance measures for the entire network are calculated. The throughput is given by

$$d = \sum_{i=1}^{n} d_i. \tag{3.20}$$

The total rate of service completions is

$$s = \sum_{j=1}^{n} s_j = \sum_{j=1}^{n} \lambda_j \tag{3.21}$$

where the rate of service completions, s_j, at node j is equivalent to the arrival rate, λ_j, at node j.

The overall steady-state mean time spent in the network is based on an entity that visits every node in the network,

$$W = W_1 + W_2 + \cdots + W_n. \tag{3.22}$$

The other three overall steady-state performance measures for the network are found in a similar manner:

$$W_q = W_{q_1} + W_{q_2} + \cdots + W_{q_n}, \tag{3.23}$$

$$L = L_1 + L_2 + \cdots + L_n, \tag{3.24}$$

$$L_q = L_{q_1} + L_{q_2} + \cdots + L_{q_n}. \tag{3.25}$$

This concludes the review of general results for a single-class model. Next, the equations and procedures specific to each of the four queueing network model scenarios are described.

3.2.1 Low Resolution, Single-Class Model

The low resolution, single-class queueing network model consists of four tandem nodes, one for each JRSOI stage. Flow is directed sequentially and once an entity enters the network it remains in the network until it completes service at the integration node. No distinction is made between pax and cargo entities. Each node has a single server and operates according to the first come, first served (FCFS) queueing discipline. Entities are neither created nor combined at the nodes, meaning bulk arrivals and departures are not modeled. The routing matrix R is

$$R = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

The simplistic structure of the routing matrix is due to the tandem configuration of the feed-forward network of Figure 3.1. Using Equation (3.1), the computed arrival rates to each node are thus

$$\begin{aligned}
\lambda_1 &= \lambda_{0,1}, \\
\lambda_2 &= \lambda_{0,2} + \lambda_1 = \lambda_{0,2} + \lambda_{0,1}, \\
\lambda_3 &= \lambda_{0,3} + \lambda_2 = \lambda_{0,3} + \lambda_{0,2} + \lambda_{0,1}, \\
\lambda_4 &= \lambda_{0,4} + \lambda_3 = \lambda_{0,4} + \lambda_{0,3} + \lambda_{0,2} + \lambda_{0,1}.
\end{aligned}$$

This enables calculation of the traffic intensities. Since $m_j = 1$ for all j,

$$\rho_j = \lambda_j / \mu_j, \qquad j = 1, 2, 3, 4.$$

The arrival rates to node j from node i are found using Equation (3.4)

$$\begin{aligned}
\lambda_{1,2} &= \lambda_1 r_{1,2} = \lambda_1 = \lambda_{0,1}, \\
\lambda_{2,3} &= \lambda_2 r_{2,3} = \lambda_2 = \lambda_{0,2} + \lambda_{0,1}, \\
\lambda_{3,4} &= \lambda_3 r_{3,4} = \lambda_3 = \lambda_{0,3} + \lambda_{0,2} + \lambda_{0,1}, \\
\lambda_{4,0} &= \lambda_4 r_{4,0} = \lambda_4 = \lambda_{0,4} + \lambda_{0,3} + \lambda_{0,2} + \lambda_{0,1},
\end{aligned}$$

where $\lambda_{i,j} = 0$ otherwise.

The proportion of node j arrivals from node i is found using Equation (3.5) in the following manner:

$$\begin{aligned}
p_{0,1} &= \lambda_{0,1}/\lambda_1 = \lambda_{0,1}/\lambda_{0,1} = 1, \\
p_{0,2} &= \lambda_{0,2}/\lambda_2 = \lambda_{0,2}/\left(\lambda_{0,2} + \lambda_{0,1}\right), \\
p_{0,3} &= \lambda_{0,3}/\lambda_3 = \lambda_{0,3}/\left(\lambda_{0,3} + \lambda_{0,2} + \lambda_{0,1}\right), \\
p_{0,4} &= \lambda_{0,4}/\lambda_4 = \lambda_{0,4}/\left(\lambda_{0,4} + \lambda_{0,3} + \lambda_{0,2} + \lambda_{0,1}\right), \\
p_{1,2} &= \lambda_{1,2}/\lambda_2 = \lambda_{0,1}/\left(\lambda_{0,2} + \lambda_{0,1}\right), \\
p_{2,3} &= \lambda_{2,3}/\lambda_3 = \left(\lambda_{0,2} + \lambda_{0,1}\right)/\left(\lambda_{0,3} + \lambda_{0,2} + \lambda_{0,1}\right), \\
p_{3,4} &= \lambda_{3,4}/\lambda_4 = \left(\lambda_{0,3} + \lambda_{0,2} + \lambda_{0,1}\right)/\left(\lambda_{0,4} + \lambda_{0,3} + \lambda_{0,2} + \lambda_{0,1}\right),
\end{aligned}$$

where $p_{i,j} = 0$ otherwise.

The departure rates out of the network from the nodes are calculated according to Equation (3.6)

$$\begin{aligned}
d_j &= 0, \qquad j = 1, 2, 3, \\
d_4 &= \lambda_4 = \sum_{j=1}^{4} \lambda_{0,j}.
\end{aligned}$$

The approximated squared coefficients of variation for the arrival processes to each node are found using Equation (3.7)

$$\begin{aligned}
c_{a1}^2 &= a_1 + \sum_{i=1}^{4} c_{ai}^2 b_{i,1} = a_1, \\
c_{a2}^2 &= a_2 + \sum_{i=1}^{4} c_{ai}^2 b_{i,2} = a_2 + c_{a1}^2 b_{1,2}, \\
c_{a3}^2 &= a_3 + \sum_{i=1}^{4} c_{ai}^2 b_{i,3} = a_3 + c_{a2}^2 b_{2,3}, \\
c_{a4}^2 &= a_4 + \sum_{i=1}^{4} c_{ai}^2 b_{i,4} = a_4 + c_{a3}^2 b_{3,4},
\end{aligned}$$

where the variability constants a_j and $b_{i,j}$ are calculated using Equations (3.8) and (3.9), respectively

$$\begin{aligned}
a_1 &= 1 + w_1 \left(c_{01}^2 - 1\right), \\
a_2 &= 1 + w_2 \left\{ \left[\left(\tfrac{\lambda_{0,2}}{\lambda_{0,2}+\lambda_{0,1}} \right) c_{0,2}^2 - 1 \right] + \left(\tfrac{\lambda_{0,1}}{\lambda_{0,2}+\lambda_{0,1}} \right) \left(\tfrac{\lambda_{0,1}}{\mu_1} \right)^2 x_1 \right\}, \\
a_3 &= 1 + w_3 \left\{ \left[\left(\tfrac{\lambda_{0,3}}{\lambda_{0,3}+\lambda_{0,2}+\lambda_{0,1}} \right) c_{03}^2 - 1 \right] + \left(\tfrac{\lambda_{0,2}+\lambda_{0,1}}{\lambda_{0,3}+\lambda_{0,2}+\lambda_{0,1}} \right) \left(\tfrac{\lambda_{0,2}+\lambda_{0,1}}{\mu_2} \right)^2 x_2 \right\}, \\
a_4 &= 1 + w_4 \left\{ \left[\left(\tfrac{\lambda_{0,4}}{\lambda_{0,4}+\lambda_{0,3}+\lambda_{0,2}+\lambda_{0,1}} \right) c_{03}^2 - 1 \right] + \left(\tfrac{\lambda_{0,3}+\lambda_{0,2}+\lambda_{0,1}}{\lambda_{0,4}+\lambda_{0,3}+\lambda_{0,2}+\lambda_{0,1}} \right) \left(\tfrac{\lambda_{0,3}+\lambda_{0,2}+\lambda_{0,1}}{\mu_3} \right)^2 x_3 \right\}, \\
a_j &= 0, \qquad \text{otherwise,}
\end{aligned}$$

and

$$\begin{aligned}
b_{1,2} &= w_2 \left[\frac{\lambda_{0,1}}{\lambda_{0,2} + \lambda_{0,1}} \right] \left[1 - \left(\frac{\lambda_{0,1}}{\mu_1} \right)^2 \right], \\
b_{2,3} &= w_3 \left[\frac{\lambda_{0,2} + \lambda_{0,1}}{\lambda_{0,3} + \lambda_{0,2} + \lambda_{0,1}} \right] \left[1 - \left(\frac{\lambda_{0,2} + \lambda_{0,1}}{\mu_2} \right)^2 \right], \\
b_{3,4} &= w_4 \left[\frac{\lambda_{0,3} + \lambda_{0,2} + \lambda_{0,1}}{\lambda_{0,4} + \lambda_{0,3} + \lambda_{0,2} + \lambda_{0,1}} \right] \left[1 - \left(\frac{\lambda_{0,3} + \lambda_{0,2} + \lambda_{0,1}}{\mu_3} \right)^2 \right], \\
b_{i,j} &= 0, \qquad \text{otherwise.}
\end{aligned}$$

The values for the variables x_j, w_j, and v_j, based on Equations (3.10), (3.11), and (3.12), respectively, are

$$x_j = \max\left\{c_{sj}^2, 0.2\right\}, \qquad j = 1, 2, 3, 4,$$

$$\begin{aligned}
w_1 &= 1,\\
w_2 &= \left\{1 + 4\left[1 - \frac{(\lambda_{0,2} + \lambda_{0,1})}{\mu_2}\right]^2 \left[\frac{(\lambda_{0,2} + \lambda_{0,1})^2}{\left(\lambda_{0,2}^2 + \lambda_{0,1}^2\right)} - 1\right]\right\}^{-1},\\
w_3 &= \left\{1 + 4\left[1 - \frac{(\lambda_{0,3} + \lambda_{0,2} + \lambda_{0,1})}{\mu_3}\right]^2 \left[\frac{(\lambda_{0,3} + \lambda_{0,2} + \lambda_{0,1})^2}{(\lambda_{0,2} + \lambda_{0,1})^2 + \lambda_{0,3}^2} - 1\right]\right\}^{-1},\\
w_4 &= \left\{1 + 4\left[1 - \frac{(\lambda_{0,4} + \lambda_{0,3} + \lambda_{0,2} + \lambda_{0,1})}{\mu_4}\right]^2 \left[\frac{(\lambda_{0,4} + \lambda_{0,3} + \lambda_{0,2} + \lambda_{0,1})^2}{(\lambda_{0,3} + \lambda_{0,2} + \lambda_{0,1})^2 + \lambda_{0,4}^2} - 1\right]\right\}^{-1},
\end{aligned}$$

and

$$\begin{aligned}
v_1 &= 1,\\
v_2 &= \frac{(\lambda_{0,2} + \lambda_{0,1})^2}{\lambda_{0,2}^2 + \lambda_{0,1}^2},\\
v_3 &= \frac{(\lambda_{0,3} + \lambda_{0,2} + \lambda_{0,1})^2}{\lambda_{0,3}^2 + (\lambda_{0,2} + \lambda_{0,1})^2},\\
v_4 &= \frac{(\lambda_{0,4} + \lambda_{0,3} + \lambda_{0,2} + \lambda_{0,1})^2}{\lambda_{0,4}^2 + (\lambda_{0,3} + \lambda_{0,2} + \lambda_{0,1})^2}.
\end{aligned}$$

The total throughput and total rate of service for the network, given by Equations (3.20) and (3.21) respectively, are

$$d = \lambda_{0,4} + \lambda_{0,3} + \lambda_{0,2} + \lambda_{0,1}, \tag{3.26}$$

and

$$s = \lambda_{0,4} + 2\lambda_{0,3} + 3\lambda_{0,2} + 4\lambda_{0,1}. \tag{3.27}$$

The remaining performance measures of interest for each node and the network as a whole can be calculated using Equations (3.13)-(3.19) and (3.22)-(3.25).

3.2.2 Low Resolution, Multi-Class Model

The network topology for the low resolution, multi-class queueing network model is identical to that of the low resolution, single-class model. However, this model incorporates four classes of entities: pax, outsize cargo, oversize cargo, and bulk cargo. Since each class can enter the network at any of the four nodes, each class is considered to have four distinct routes depending upon where it entered the network. Therefore, there are 16 class/route combinations. The reception, staging, onward movement, and integration nodes are denoted as nodes 1, 2, 3, and 4 respectively. Pax, outsize cargo, oversize cargo, and bulk cargo class/routes are denoted by the subscripts Pi, CTi, CVi, and CBi respectively where $i = 1, 2, 3, 4$ indicates the route starting at node i. It is assumed that the routing matrix is state-independent. Each node operates according to the first come, first served (FCFS) queueing discipline and has a single server; however, the service rate is dependent upon the class/route. Entities are neither created nor combined at the nodes, meaning bulk arrivals and departures are not modeled.

For models with multiple classes, the number of nodes on class/route k, denoted by n_k, must be determined. Based on the information above, the n_k for this model configuration are

$$n_{Pj} = n_{CTj} = n_{CVj} = n_{CBj} = 5 - j, \qquad j = 1, 2, 3, 4. \tag{3.28}$$

This simply indicates that entities entering the network at the reception node have four nodes on their route, entities entering the network at the staging node have three nodes on their route, etc. Let $n_{k,j}$ indicate the jth node visited by class/route

k. Then

$$\begin{array}{llll} n_{K1,1} = 1 & n_{K1,2} = 2 & n_{K1,3} = 3 & n_{K1,4} = 4 \\ n_{K2,1} = 2 & n_{K2,2} = 3 & n_{K2,3} = 4 & \\ n_{K3,1} = 3 & n_{K3,2} = 4 & & \\ n_{K4,1} = 4, & & & \end{array}$$

where $K \in \{P, CT, CV, CB\}$. As an example $n_{CV3,2} = 4$ indicates that the second node on the route for class/route $CV3$ is the integration node (node 4).

The external arrival rates for class/route k are denoted by $\hat{\lambda}_k$. These are combined to form aggregate external arrival rates for each node using the following equation:

$$\lambda_{0,j} = \sum_{k=1}^{r} \hat{\lambda}_k I_{\{k:n_{k,1}=j\}}, \qquad (3.29)$$

where I is the indicator function (i.e., $I_K = 1$ when $k \in K$ and $I_K = 0$ otherwise) and r is the number of possible class/routes. This means the external arrival rate to node j is the sum of all external arrival rates of class/routes that have node j as the first node in their route. Therefore, the external arrival rates are

$$\lambda_{0,j} = \hat{\lambda}_{Pj} + \hat{\lambda}_{CTj} + \hat{\lambda}_{CVj} + \hat{\lambda}_{CBj}, \qquad j = 1, 2, 3, 4.$$

Similarly, the rates of flow from node i to node j are obtained by using the following equation where k represents the class/route:

$$\lambda_{i,j} = \sum_{k=1}^{r} \sum_{l=1}^{n_k - 1} \hat{\lambda}_k I_{\{(k,l):n_{k,l}=i, n_{k,l+1}=j\}}, \qquad (3.30)$$

This function merely sums the external arrival rates $\hat{\lambda}_k$ for all those class/routes whose route includes a direct flow from node i to node j. (If a class/route k includes m such direct flows from node i to node j then $\hat{\lambda}_k$ is added m times when calculating

$\lambda_{i,j}$.) The rates for this model configuration are as follows:

$$\begin{aligned}
\lambda_{1,2} &= \sum_{i=1}^{1} \left(\hat{\lambda}_{Pi} + \hat{\lambda}_{CTi} + \hat{\lambda}_{CVi} + \hat{\lambda}_{CBi}\right) = \lambda_{0,1}, \\
\lambda_{2,3} &= \sum_{i=1}^{2} \left(\hat{\lambda}_{Pi} + \hat{\lambda}_{CTi} + \hat{\lambda}_{CVi} + \hat{\lambda}_{CBi}\right) = \lambda_{0,1} + \lambda_{0,2}, \\
\lambda_{3,4} &= \sum_{i=1}^{3} \left(\hat{\lambda}_{Pi} + \hat{\lambda}_{CTi} + \hat{\lambda}_{CVi} + \hat{\lambda}_{CBi}\right) = \lambda_{0,1} + \lambda_{0,2} + \lambda_{0,3},
\end{aligned}$$

where $\lambda_{i,j} = 0$ otherwise. The departure rate from the network out of node i is

$$d_i = \lambda_{i,0} = \sum_{k=1}^{r} \hat{\lambda}_k I_{\{k:n_{k,n_k}=i\}}, \tag{3.31}$$

This means the departure rate from node i is the sum of all external arrival rates of class/routes that have node i as the last node in their route. Therefore, the departure rates are

$$\begin{aligned}
d_i &= \lambda_{i,0} = 0, \qquad i = 1, 2, 3, \\
d_4 &= \lambda_{4,0} = \sum_{i=1}^{4} \left(\hat{\lambda}_{Pi} + \hat{\lambda}_{CTi} + \hat{\lambda}_{CVi} + \hat{\lambda}_{CBi}\right) = \lambda_{0,1} + \lambda_{0,2} + \lambda_{0,3} + \lambda_{0,4}.
\end{aligned}$$

The proportion of entities at node i that proceed to node j is

$$r_{i,j} = \frac{\lambda_{i,j}}{\lambda_{i,0} + \sum_{h=1}^{4} \lambda_{i,h}}. \tag{3.32}$$

By no coincidence, the R matrix turns out to be the same as for the low resolution, single-class configuration:

$$R = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Using Equation (3.1), the computed arrival rates to each node are thus

$$\begin{aligned}
\lambda_1 &= \lambda_{0,1}, \\
\lambda_2 &= \lambda_{0,2} + \lambda_{0,1}, \\
\lambda_3 &= \lambda_{0,3} + \lambda_{0,2} + \lambda_{0,1}, \\
\lambda_4 &= \lambda_{0,4} + \lambda_{0,3} + \lambda_{0,2} + \lambda_{0,1}.
\end{aligned}$$

The proportion of arrivals to node j coming from node i is found using Equation (3.5) in the following manner:

$$\begin{aligned}
p_{0,1} &= \lambda_{0,1}/\lambda_1 = \lambda_{0,1}/\lambda_{0,1} = 1, \\
p_{0,2} &= \lambda_{0,2}/\lambda_2 = \lambda_{0,2}/\left(\lambda_{0,2} + \lambda_{0,1}\right), \\
p_{0,3} &= \lambda_{0,3}/\lambda_3 = \lambda_{0,3}/\left(\lambda_{0,3}+, \lambda_{0,2} + \lambda_{0,1}\right), \\
p_{0,4} &= \lambda_{0,4}/\lambda_4 = \lambda_{0,4}/\left(\lambda_{0,4} + \lambda_{0,3} + \lambda_{0,2} + \lambda_{0,1}\right), \\
p_{1,2} &= \lambda_{1,2}/\lambda_2 = \lambda_{0,1}/\left(\lambda_{0,2} + \lambda_{0,1}\right), \\
p_{2,3} &= \lambda_{2,3}/\lambda_3 = \left(\lambda_{0,2} + \lambda_{0,1}\right)/\left(\lambda_{0,3} + \lambda_{0,2} + \lambda_{0,1}\right), \\
p_{3,4} &= \lambda_{3,4}/\lambda_4 = \left(\lambda_{0,3} + \lambda_{0,2} + \lambda_{0,1}\right)/\left(\lambda_{0,4} + \lambda_{0,3} + \lambda_{0,2} + \lambda_{0,1}\right),
\end{aligned}$$

where $p_{i,j} = 0$ otherwise.

Next, aggregate service-time parameters are found. Let $\hat{\mu}_{k,l}$ be the service rate at the lth node of route k. Then the aggregate service rates at each node are found by averaging the class/route specific service rates:

$$\mu_j = \frac{\sum_{k=1}^{r} \sum_{l=1}^{n_k} \hat{\lambda}_k I_{\{(k,l):n_{k,l}=j\}}}{\sum_{k=1}^{r} \sum_{l=1}^{n_k} \left(\hat{\lambda}_k/\hat{\mu}_{k,l}\right) I_{\{(k,l):n_{k,l}=j\}}}, \tag{3.33}$$

For each node j, the numerator sums the external arrival rates for the class/routes that include j somewhere in their route. If a node is visited m times on a given route, then the external arrival rate for that class/route is included m times in the sum. The denominator is equivalent to the numerator with the exception that each summand is divided by the service rate at node j according to where j occurs on the corresponding class/route. The service rates for this model configuration are thus

$$\mu_1 = \frac{\hat{\lambda}_{P1} + \hat{\lambda}_{CT1} + \hat{\lambda}_{CV1} + \hat{\lambda}_{CB1}}{\sum_K \frac{\hat{\lambda}_{K1}}{\hat{\mu}_{K1,1}}},$$

$$\mu_2 = \frac{\sum_{i=1}^{2} \left(\hat{\lambda}_{Pi} + \hat{\lambda}_{CTi} + \hat{\lambda}_{CVi} + \hat{\lambda}_{CBi}\right)}{\sum_K \left(\frac{\hat{\lambda}_{K1}}{\hat{\mu}_{K1,2}} + \frac{\hat{\lambda}_{K2}}{\hat{\mu}_{K2,1}}\right)},$$

$$\mu_3 = \frac{\sum_{i=1}^{3} \left(\hat{\lambda}_{Pi} + \hat{\lambda}_{CTi} + \hat{\lambda}_{CVi} + \hat{\lambda}_{CBi}\right)}{\sum_K \left(\frac{\hat{\lambda}_{K1}}{\hat{\mu}_{K1,3}} + \frac{\hat{\lambda}_{K2}}{\hat{\mu}_{K2,2}} + \frac{\hat{\lambda}_{K3}}{\hat{\mu}_{K3,1}}\right)},$$

$$\mu_4 = \frac{\sum_{i=1}^{4} \left(\hat{\lambda}_{Pi} + \hat{\lambda}_{CTi} + \hat{\lambda}_{CVi} + \hat{\lambda}_{CBi}\right)}{\sum_K \left(\frac{\hat{\lambda}_{K1}}{\hat{\mu}_{K1,4}} + \frac{\hat{\lambda}_{K2}}{\hat{\mu}_{K2,3}} + \frac{\hat{\lambda}_{K3}}{\hat{\mu}_{K3,2}} + \frac{\hat{\lambda}_{K4}}{\hat{\mu}_{K4,1}}\right)},$$

where $K \in \{P, CT, CV, CB\}$. The traffic intensity ρ_j can now be found for each node by

$$\rho_j = \lambda_j/\mu_j, \qquad j = 1, 2, 3, 4.$$

Let $\hat{c}^2_{sk,l}$ be the service squared coefficient of variation at the lth node of route k. Then the aggregate service squared coefficients of variation c^2_{sj} are found using the following equation which is similar in construction to the one used for calculating the aggregate service rates:

$$c^2_{sj} = \left\{ \frac{\sum_{k=1}^{r} \sum_{l=1}^{n_k} \left(\hat{\lambda}_k/\hat{\mu}^2_{k,l}\right) \left(\hat{c}^2_{sk,l} + 1\right) I_{\{(k,l):n_{k,l}=j\}}}{\sum_{k=1}^{r} \sum_{l=1}^{n_k} \hat{\lambda}_k I_{\{(k,l):n_{k,l}=j\}}} \right\} - 1. \tag{3.34}$$

The aggregate service squared coefficients of variation for this model configuration are

$$c_{s1}^2 = \left\{ \frac{\mu_1^2 \sum_K \frac{\hat{\lambda}_{K1}\left(\hat{c}_{sK1,1}^2+1\right)}{\hat{\mu}_{K1,1}^2}}{\hat{\lambda}_{P1}+\hat{\lambda}_{CT1}+\hat{\lambda}_{CV1}+\hat{\lambda}_{CB1}} \right\} - 1,$$

$$c_{s2}^2 = \left\{ \frac{\mu_2^2 \sum_K \left(\frac{\hat{\lambda}_{K1}\left(\hat{c}_{sK1,2}^2+1\right)}{\hat{\mu}_{K1,2}^2} + \frac{\hat{\lambda}_{K2}\left(\hat{c}_{sK2,2}^2+1\right)}{\hat{\mu}_{K2,1}^2} \right)}{\sum_{i=1}^2 \left(\hat{\lambda}_{Pi}+\hat{\lambda}_{CTi}+\hat{\lambda}_{CVi}+\hat{\lambda}_{CBi}\right)} \right\} - 1,$$

$$c_{s3}^2 = \left\{ \frac{\mu_3^2 \sum_K \left(\frac{\hat{\lambda}_{K1}\left(\hat{c}_{sK1,3}^2+1\right)}{\hat{\mu}_{K1,3}^2} + \frac{\hat{\lambda}_{K2}\left(\hat{c}_{sK2,2}^2+1\right)}{\hat{\mu}_{K2,2}^2} + \frac{\hat{\lambda}_{K3}\left(\hat{c}_{sK3,1}^2+1\right)}{\hat{\mu}_{K3,1}^2} \right)}{\sum_{i=1}^3 \left(\hat{\lambda}_{Pi}+\hat{\lambda}_{CTi}+\hat{\lambda}_{CVi}+\hat{\lambda}_{CBi}\right)} \right\} - 1,$$

$$c_{s4}^2 = \left\{ \frac{\mu_4^2 \sum_K \left(\frac{\hat{\lambda}_{K1}\left(\hat{c}_{sK1,4}^2+1\right)}{\hat{\mu}_{K1,4}^2} + \frac{\hat{\lambda}_{K2}\left(\hat{c}_{sK2,3}^2+1\right)}{\hat{\mu}_{K2,3}^2} + \frac{\hat{\lambda}_{K3}\left(\hat{c}_{sK3,2}^2+1\right)}{\hat{\mu}_{K3,2}^2} + \frac{\hat{\lambda}_{K4}\left(\hat{c}_{sK4,1}^2+1\right)}{\hat{\mu}_{K4,1}^2} \right)}{\sum_{i=1}^4 \left(\hat{\lambda}_{Pi}+\hat{\lambda}_{CTi}+\hat{\lambda}_{CVi}+\hat{\lambda}_{CBi}\right)} \right\} - 1.$$

Let $\hat{c}_{0k}^2$ be the arrival squared coefficient of variation for route k. Then the aggregate external arrival square coefficients of variation are found using the following equation:

$$c_{0j}^2 = (1-\hat{w}_j) + \hat{w}_j \left\{ \sum_{k=1}^{r} \hat{c}_{0k}^2 \left(\frac{\hat{\lambda}_k I_{\{k:n_{k,1}=j\}}}{\sum_{l=1}^{r} \hat{\lambda}_l I_{\{l:n_{l,1}=j\}}} \right) \right\}, \qquad j = 1,2,3,4 \tag{3.35}$$

where the $\hat{w}_j$ are defined as

$$\hat{w}_j = \left\{ 1 + 4\left(1-\rho_j\right)^2 \left(\hat{v}_j - 1\right) \right\}^{-1}, \tag{3.36}$$

and the $\hat{v}_j$ are defined as

$$\hat{v}_j = \left\{ \sum_{k=1}^{r} \left(\frac{\hat{\lambda}_k I_{\{k:n_{k,1}=j\}}}{\sum_{l=1}^{r} \hat{\lambda}_l I_{\{l:n_{l,1}=j\}}} \right)^2 \right\}^{-1}, \qquad j = 1,2,3,4. \tag{3.37}$$

Thus, for this model configuration, the aggregate external arrival squared coefficients of variation are

$$c_{0j}^2 = (1-\hat{w}_j) + \hat{w}_j \left\{ \sum_K \hat{c}_{0Kj}^2 \left(\frac{\hat{\lambda}_{Kj}}{\hat{\lambda}_{Pj} + \hat{\lambda}_{CTj} + \hat{\lambda}_{CVj} + \hat{\lambda}_{CBj}} \right) \right\},$$

where the $\hat{w}_j$ can be found using Equation (3.36) and the $\hat{v}_j$ are

$$\hat{v}_j = \left\{ \sum_K \left(\frac{\hat{\lambda}_{Kj}}{\hat{\lambda}_{Pj} + \hat{\lambda}_{CTj} + \hat{\lambda}_{CVj} + \hat{\lambda}_{CBj}} \right)^2 \right\}^{-1} = \left\{ \sum_K \left(\frac{\hat{\lambda}_{Kj}}{\hat{\lambda}_j} \right)^2 \right\}^{-1},$$

where $K \in \{P, CT, CV, CB\}$ and $j = 1, 2, 3, 4$.

Now that the c_{0j} have been found, the remaining equations can be worked out. These are identical to those of the low resolution, single-class configuration starting with the equations to approximate the squared coefficients of variation for the arrival processes at each node (Equations (3.7) to (3.25)).

3.2.3 High Resolution, Single-Class Model

The high resolution, single-class queueing network model consists of twenty-one nodes comprising four subnetworks representing the four stages of JRSOI (See Table 3.1). This model is notional and is based on information from multiple sources including JP4-01.8 [9], the Defense Transportation Regulation [13], and subject matter experts. It was designed to capture the salient features and network constructs of the generalized JRSOI process, not to model the specific network for every situation and operating location. Real-world JRSOI networks are classified and their configurations are situation- and operating-location specific. In practice, QNA can analyze the network configuration specific to each situation and operating location to a desired level of resolution.

The model includes tandem constructs, probabilistic branching, and a fork-join node. Flow is directed sequentially, and once an entity enters the network, it remains

Table 3.1 Network node descriptions.

Node	Description
1	Offload pax and cargo at APOD
2	Clear pax and cargo
3	Move pax and cargo to holding area
4	In-process pax (reception phase)
5	Inspect vehicles and perform maintenance
6	Breakdown and reorganize pallets
7	Hold sensitive cargo
8	Move pax and cargo to marshalling area
9	Marshall pax and cargo
10	Move pax and cargo to staging area
11	Receive and inspect cargo
12	In-process pax (staging phase)
13	Conduct operability checks on cargo
14	Conduct training of pax as required
15	Prepare pax and cargo for onward movement
16	Plane
17	Train
18	Convoy
19	Receive and inspect cargo
20	In-process pax (integration phase)
21	Conduct integration

in the network until it completes service at the integration node. Entities enter the network at node 1 or node 10. No distinction is made between pax and cargo entities. Each node has a single server and operates according to the first come, first served (FCFS) queueing discipline. Bulk arrivals and departures are not considered.

One of the unique features of this model is the incorporation of a fork-join queueing station. A fork-join queueing station is one where, upon entering into service, an entity awaits multiple concurrent service activities. Therefore, service for that entity is completed upon completion of the service process taking the longest amount of time to complete. In a stochastic environment, this will not necessarily be the same process every time.

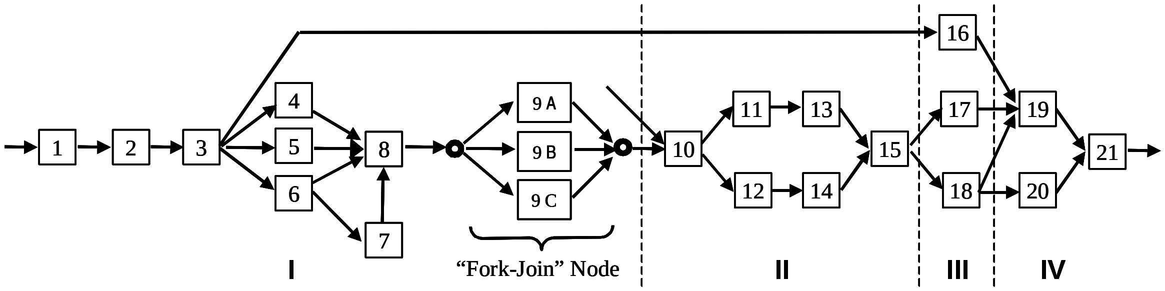

Figure 3.2 Graphical depiction of a high resolution JRSOI network.

Based on assigned routing probabilities, the routing matrix R is

$$R = \begin{pmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & .4 & .2 & .3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .1 & .9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .7 & .3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .4 & .6 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .7 & .3 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}.$$

The simplistic structure of the routing matrix is due to the nature of the model's feed-forward network. Using Equation (3.1), the computed arrival rates are thus

$$\begin{aligned}
\lambda_1 &= \lambda_2 = \lambda_3 = \lambda_{0,1}, \\
\lambda_4 &= .4\lambda_3 = .4\lambda_{0,1}, \\
\lambda_5 &= .2\lambda_3 = .2\lambda_{0,1}, \\
\lambda_6 &= .3\lambda_3 = .3\lambda_{0,1}, \\
\lambda_7 &= .1\lambda_6 = .03\lambda_{0,1}, \\
\lambda_8 &= \lambda_4 + \lambda_5 + .9\lambda_6 + \lambda_7 = .9\lambda_{0,1}, \\
\lambda_9 &= \lambda_8 = .9\lambda_{0,1}, \\
\lambda_{10} &= \lambda_{0,10} + \lambda_9 = \lambda_{0,10} + .9\lambda_{0,1}, \\
\lambda_{11} &= .7\lambda_{10} = .7\lambda_{0,10} + .63\lambda_{0,1}, \\
\lambda_{12} &= .3\lambda_{10} = .3\lambda_{0,10} + .27\lambda_{0,1}, \\
\lambda_{13} &= \lambda_{11} = .7\lambda_{0,10} + .63\lambda_{0,1}, \\
\lambda_{14} &= \lambda_{12} = .3\lambda_{0,10} + .27\lambda_{0,1}, \\
\lambda_{15} &= \lambda_{13} + \lambda_{14} = \lambda_{0,10} + .9\lambda_{0,1}, \\
\lambda_{16} &= .1\lambda_3 = .1\lambda_{0,1}, \\
\lambda_{17} &= .4\lambda_{15} = .4\lambda_{0,10} + .36\lambda_{0,1}, \\
\lambda_{18} &= .6\lambda_{15} = .6\lambda_{0,10} + .54\lambda_{0,1}, \\
\lambda_{19} &= \lambda_{16} + \lambda_{17} + .7\lambda_{18} = .82\lambda_{0,10} + .838\lambda_{0,1}, \\
\lambda_{20} &= .3\lambda_{18} = .18\lambda_{0,10} + .162\lambda_{0,1}, \\
\lambda_{21} &= \lambda_{19} + \lambda_{20} = \lambda_{0,10} + \lambda_{0,1}.
\end{aligned}$$

This enables calculation of the traffic intensities. Since $m_j = 1$ for all j,

$$\rho_j = \lambda_j / \mu_j, \qquad j = 1, 2, ..., 21.$$

The arrival rates to node j from node i are found using Equation (3.4)

$$\begin{aligned}
\lambda_{1,2} &= \lambda_1 r_{1,2} = \lambda_1 = \lambda_{0,1},\\
\lambda_{2,3} &= \lambda_2 r_{2,3} = \lambda_2 = \lambda_{0,1},\\
\lambda_{3,4} &= \lambda_3 r_{3,4} = .4\lambda_3 = .4\lambda_{0,1},\\
\lambda_{3,5} &= \lambda_3 r_{3,5} = .2\lambda_3 = .2\lambda_{0,1},\\
\lambda_{3,6} &= \lambda_3 r_{3,6} = .3\lambda_3 = .3\lambda_{0,1},\\
\lambda_{3,16} &= \lambda_3 r_{3,16} = .1\lambda_3 = .1\lambda_{0,1},\\
\lambda_{4,8} &= \lambda_4 r_{4,8} = \lambda_4 = .4\lambda_{0,1},\\
\lambda_{5,8} &= \lambda_5 r_{5,8} = \lambda_5 = .2\lambda_{0,1},\\
\lambda_{6,7} &= \lambda_6 r_{6,7} = .1\lambda_6 = .03\lambda_{0,1},\\
\lambda_{6,8} &= \lambda_6 r_{6,8} = .9\lambda_6 = .27\lambda_{0,1},\\
\lambda_{7,8} &= \lambda_7 r_{7,8} = \lambda_7 = .03\lambda_{0,1},\\
\lambda_{8,9} &= \lambda_8 r_{8,9} = \lambda_8 = .9\lambda_{0,1},\\
\lambda_{9,10} &= \lambda_9 r_{9,10} = \lambda_9 = .9\lambda_{0,1},\\
\lambda_{10,11} &= \lambda_9 r_{10,11} = .7\lambda_{10} = .7\lambda_{0,10} + .63\lambda_{0,1},\\
\lambda_{10,12} &= \lambda_{10} r_{10,12} = .3\lambda_{10} = .3\lambda_{0,10} + .27\lambda_{0,1},\\
\lambda_{11,13} &= \lambda_{11} r_{11,13} = \lambda_{11} = .7\lambda_{0,10} + .63\lambda_{0,1},\\
\lambda_{12,14} &= \lambda_{12} r_{12,14} = \lambda_{12} = .3\lambda_{0,10} + .27\lambda_{0,1},\\
\lambda_{13,15} &= \lambda_{13} r_{13,15} = \lambda_{13} = .7\lambda_{0,10} + .63\lambda_{0,1},\\
\lambda_{14,15} &= \lambda_{14} r_{14,15} = \lambda_{14} = .3\lambda_{0,10} + .27\lambda_{0,1},\\
\lambda_{15,17} &= \lambda_{15} r_{15,17} = .4\lambda_{15} = .4\lambda_{0,10} + .36\lambda_{0,1},\\
\lambda_{15,18} &= \lambda_{15} r_{15,18} = .6\lambda_{15} = .6\lambda_{0,10} + .54\lambda_{0,1},\\
\lambda_{16,19} &= \lambda_{16} r_{16,19} = \lambda_{16} = .1\lambda_{0,1},\\
\lambda_{17,19} &= \lambda_{17} r_{17,19} = \lambda_{17} = .4\lambda_{0,10} + .36\lambda_{0,1},\\
\lambda_{18,19} &= \lambda_{18} r_{18,19} = .7\lambda_{18} = .42\lambda_{0,10} + .378\lambda_{0,1},\\
\lambda_{18,20} &= \lambda_{18} r_{18,20} = .3\lambda_{18} = .18\lambda_{0,10} + .162\lambda_{0,1},\\
\lambda_{19,21} &= \lambda_{19} r_{19,21} = \lambda_{19} = .82\lambda_{0,10} + .838\lambda_{0,1},\\
\lambda_{20,21} &= \lambda_{20} r_{20,21} = \lambda_{20} = .18\lambda_{0,10} + .162\lambda_{0,1},
\end{aligned}$$

where $\lambda_{i,j} = 0$ otherwise.

The proportion of node j arrivals from node i is found using Equation (3.5)

$$\begin{aligned} p_{0,1} &= \lambda_{0,1}/\lambda_1 = 1, \\ p_{0,10} &= \lambda_{0,10}/\lambda_{10} = \lambda_{0,10}/\left(\lambda_{0,10} + .9\lambda_{0,1}\right), \end{aligned}$$

where $p_{0,j} = 0$ otherwise for external arrivals and

$$\begin{aligned} p_{4,8} &= \lambda_{4,8}/\lambda_8 = .4\lambda_{0,1}/.9\lambda_{0,1} = 4/9, \\ p_{5,8} &= \lambda_{5,8}/\lambda_8 = .2\lambda_{0,1}/.9\lambda_{0,1} = 2/9, \\ p_{6,8} &= \lambda_{6,8}/\lambda_8 = .27\lambda_{0,1}/.9\lambda_{0,1} = 3/10, \\ p_{7,8} &= \lambda_{7,8}/\lambda_8 = .03\lambda_{0,1}/.9\lambda_{0,1} = 1/30, \\ p_{9,10} &= \lambda_{9,10}/\lambda_{10} = .9\lambda_{0,1}/\left(\lambda_{0,10} + .9\lambda_{0,1}\right), \\ p_{13,15} &= \lambda_{13,15}/\lambda_{15} = \left(.7\lambda_{0,10} + .63\lambda_{0,1}\right)/\left(\lambda_{0,10} + .9\lambda_{0,1}\right), \\ p_{14,15} &= \lambda_{14,15}/\lambda_{15} = \left(.3\lambda_{0,10} + .27\lambda_{0,1}\right)/\left(\lambda_{0,10} + .9\lambda_{0,1}\right), \\ p_{16,19} &= \lambda_{16,19}/\lambda_{19} = \left(.1\lambda_{0,1}\right)/\left(.82\lambda_{0,10} + .838\lambda_{0,1}\right), \\ p_{17,19} &= \lambda_{17,19}/\lambda_{19} = \left(.4\lambda_{0,10} + .36\lambda_{0,1}\right)/\left(.82\lambda_{0,10} + .838\lambda_{0,1}\right), \\ p_{18,19} &= \lambda_{18,19}/\lambda_{19} = \left(.42\lambda_{0,10} + .378\lambda_{0,1}\right)/\left(.82\lambda_{0,10} + .838\lambda_{0,1}\right), \\ p_{19,21} &= \lambda_{19,21}/\lambda_{21} = \left(.82\lambda_{0,10} + .838\lambda_{0,1}\right)/\left(\lambda_{0,10} + \lambda_{0,1}\right), \\ p_{20,21} &= \lambda_{20,21}/\lambda_{21} = \left(.18\lambda_{0,10} + .162\lambda_{0,1}\right)/\left(\lambda_{0,10} + \lambda_{0,1}\right), \end{aligned}$$

where $p_{i,j} = 1$ for $(i,j) \in \{(1,2),(2,3),(3,4),(3,5),(3,6),(3,16),(6,7),(8,9),(10,11),$ $(10,12),(11,13),(12,14),(15,17),(15,18),(16,19),(18,20)\}$ and $p_{i,j} = 0$ otherwise.

The departure rates out of the network from the nodes are, by Equation (3.6),

$$\begin{aligned} d_i &= 0, \qquad i = 1,2,...,20 \\ d_{21} &= \lambda_{21} = \sum_{i=1}^{21} \lambda_{0,i} = \lambda_{0,10} + \lambda_{0,1}. \end{aligned}$$

The approximated squared coefficients of variation for the arrival processes to each node are found using Equation (3.7)

$$
\begin{aligned}
c_{a1}^2 &= a_1 + \sum_{i=1}^{21} c_{ai}^2 b_{i,1} = a_1, \\
c_{a2}^2 &= a_2 + \sum_{i=1}^{21} c_{ai}^2 b_{i,2} = a_2 + c_{a1}^2 b_{1,2}, \\
c_{a3}^2 &= a_3 + \sum_{i=1}^{21} c_{ai}^2 b_{i,3} = a_3 + c_{a2}^2 b_{2,3}, \\
c_{a4}^2 &= a_4 + c_{a3}^2 b_{3,4}, \\
c_{a5}^2 &= a_5 + c_{a3}^2 b_{3,5}, \\
c_{a6}^2 &= a_6 + c_{a3}^2 b_{3,6}, \\
c_{a7}^2 &= a_7 + c_{a6}^2 b_{6,7}, \\
c_{a8}^2 &= a_8 + c_{a4}^2 b_{4,8} + c_{a5}^2 b_{5,8} + c_{a6}^2 b_{6,8} + c_{a7}^2 b_{7,8}, \\
c_{a9}^2 &= a_9 + c_{a8}^2 b_{8,9}, \\
c_{a10}^2 &= a_{10} + c_{a9}^2 b_{9,10}, \\
c_{a11}^2 &= a_{11} + c_{a10}^2 b_{10,11}, \\
c_{a12}^2 &= a_{12} + c_{a10}^2 b_{10,12}, \\
c_{a13}^2 &= a_{13} + c_{a11}^2 b_{11,13}, \\
c_{a14}^2 &= a_{14} + c_{a12}^2 b_{12,14}, \\
c_{a15}^2 &= a_{15} + c_{a13}^2 b_{13,15} + c_{a14}^2 b_{14,15}, \\
c_{a16}^2 &= a_{16} + c_{a3}^2 b_{3,16}, \\
c_{a17}^2 &= a_{17} + c_{a15}^2 b_{15,17}, \\
c_{a18}^2 &= a_{18} + c_{a15}^2 b_{15,18}, \\
c_{a19}^2 &= a_{19} + c_{a16}^2 b_{16,19} + c_{a17}^2 b_{17,19} + c_{a18}^2 b_{18,19}, \\
c_{a20}^2 &= a_{20} + c_{a18}^2 b_{18,20}, \\
c_{a21}^2 &= a_{21} + c_{a19}^2 b_{19,21} + c_{a20}^2 b_{20,21},
\end{aligned}
$$

where the variability constants a_j and $b_{i,j}$ are calculated using Equations (3.8) and (3.9), respectively

$$a_j = 1 + w_j \left\{ \left(p_{0,j} c_{0j}^2 - 1 \right) + \sum_{i=1}^{n} p_{i,j} \left[(1 - r_{i,j}) + r_{i,j} \rho_i^2 x_i \right] \right\}$$

and

$$b_{i,j} = w_j p_{i,j} r_{i,j} \left(1 - \rho_i^2 \right).$$

The values for the variables x_j, w_j, and v_j, based on Equations (3.10), (3.11), and (3.12), respectively, are

$$x_j = \max \left\{ c_{sj}^2, 0.2 \right\}, \qquad j = 1, 2, 3, 4$$

and

$$w_j = \left[1 + 4 \left(1 - \rho_j \right)^2 \left(v_j - 1 \right) \right]^{-1},$$

with

$$v_j = \left(\sum_{i=0}^{21} p_{i,j}^2 \right)^{-1}.$$

The total throughput for the network is given by

$$d = \lambda_{0,10} + \lambda_{0,1}$$

and the total rate of service completions is

$$s = 7\lambda_{0,10} + 10.33\lambda_{0,1}.$$

The remaining performance measures of interest for each node and the network as a whole can be calculated using Equations (3.13)-(3.19) and (3.22)-(3.25).

3.2.4 High Resolution, Multi-Class Model

The network topology for the high resolution, multi-class queueing network model is identical to that of the high resolution, single-class model. However, this model incorporates the four entity classes: pax, outsize cargo, oversize cargo, and bulk cargo. Each possible route for each class, based on the probabilistic routing given for the high resolution, single-class model, forms a distinct deterministic route for this model. An additional route is added to account for combat units that enter JRSOI already intact. In total, there are 13 routes for this model. The nodes are described in Table 3.1. The routes are given in Table 3.2. It is assumed that the routing matrix is state-independent. Each node operates according to the first come, first served (FCFS) queueing discipline and has a single server; however, the service rate is dependent upon the class and route. Bulk arrivals and departures are not modeled.

Table 3.2 Enumeration of high resolution, multi-class network routes.

Route No.	Label	Path	n_k
1	Pax1	1,2,3,4,8,9,10,12,14,15,18,20,21	13
2	Pax2	10,12,14,15,18,20,21	7
3	Outsize1	1,2,3,5,8,9,10,11,13,15,17,19,21	13
4	Outsize2	10,11,13,15,18,19,21	7
5	Oversize1	1,2,3,6,8,9,10,11,13,15,17,19,21	13
6	Oversize2	1,2,3,6,7,8,9,10,11,13,15,18,19,21	14
7	Oversize3	1,2,3,5,8,9,10,11,13,15,18,19,21	13
8	Bulk1	1,2,3,6,8,9,10,11,13,15,17,19,21	13
9	Bulk2	1,2,3,6,8,9,10,11,13,15,18,19,21	13
10	Bulk3	1,2,3,6,7,8,9,10,11,13,15,17,19,21	14
11	Bulk4	1,2,3,6,7,8,9,10,11,13,15,18,19,21	14
12	Bulk5	10,11,13,15,18,20,21	7
13	Unit	1,2,3,16,19,21	6

The number of nodes on route k, denoted by n_k, are given in Table 3.2. In addition, the $n_{k,j}$ values, indicating the jth node visited by route k, can easily be determined using the information in Table 3.2.

The external arrival rates for route k are denoted by $\hat{\lambda}_k$. These are combined to form aggregate external arrival rates for each of the 21 nodes using Equation (3.29)

$$\lambda_{0,j} = \sum_{k=1}^{13} \hat{\lambda}_k I_{\{k:n_{k,1}=j\}}, \qquad j = 1, 2, ..., 21$$

where I is the indicator function and the summation is over all possible routes. This means the external arrival rate to node j is the sum of all external arrival rates of routes that have node j as their first node. Therefore, the external arrival rates are

$$\begin{aligned} \lambda_{0,1} &= \hat{\lambda}_1 + \hat{\lambda}_3 + \hat{\lambda}_5 + \hat{\lambda}_6 + \hat{\lambda}_7 + \hat{\lambda}_8 + \hat{\lambda}_9 + \hat{\lambda}_{10} + \hat{\lambda}_{11} + \hat{\lambda}_{13}, \\ \lambda_{0,10} &= \hat{\lambda}_2 + \hat{\lambda}_4 + \hat{\lambda}_{12}, \\ \lambda_{0,j} &= 0, \qquad \text{otherwise.} \end{aligned}$$

Similarly, the rates of flow from node i to node j are obtained by using Equation (3.30) where k represents the route:

$$\lambda_{ij} = \sum_{k=1}^{13} \sum_{l=1}^{n_k-1} \hat{\lambda}_k I_{\{(k,l):n_{k,l}=i,n_{k,l+1}=j\}}, \qquad i, j = 1, 2, ..., 21.$$

This function sums the external arrival rates $\hat{\lambda}_k$ for all routes that include a direct flow from node i to node j. (If route k includes m direct flows from node i to node j then $\hat{\lambda}_k$ is added m times when calculating $\lambda_{i,j}$.) The rates for this model configuration are as follows:

$$\begin{aligned}
\lambda_{1,2} &= \lambda_{2,3} = \hat{\lambda}_1 + \hat{\lambda}_3 + \hat{\lambda}_5 + \hat{\lambda}_6 + \hat{\lambda}_7 + \hat{\lambda}_8 + \hat{\lambda}_9 + \hat{\lambda}_{10} + \hat{\lambda}_{11} + \hat{\lambda}_{13}, \\
\lambda_{3,4} &= \lambda_{4,8} = \hat{\lambda}_1, \\
\lambda_{3,5} &= \lambda_{5,8} = \hat{\lambda}_3 + \hat{\lambda}_7, \\
\lambda_{3,6} &= \hat{\lambda}_5 + \hat{\lambda}_6 + \hat{\lambda}_8 + \hat{\lambda}_9 + \hat{\lambda}_{10} + \hat{\lambda}_{11}, \\
\lambda_{3,16} &= \lambda_{16,19} = \hat{\lambda}_{13}, \\
\lambda_{6,7} &= \lambda_{7,8} = \hat{\lambda}_6 + \hat{\lambda}_{10} + \hat{\lambda}_{11}, \\
\lambda_{6,8} &= \hat{\lambda}_5 + \hat{\lambda}_8 + \hat{\lambda}_9, \\
\lambda_{8,9} &= \lambda_{9,10} = \hat{\lambda}_1 + \hat{\lambda}_3 + \hat{\lambda}_5 + \hat{\lambda}_6 + \hat{\lambda}_7 + \hat{\lambda}_8 + \hat{\lambda}_9 + \hat{\lambda}_{10} + \hat{\lambda}_{11}, \\
\lambda_{10,11} &= \lambda_{11,13} = \lambda_{13,15} = \hat{\lambda}_3 + \hat{\lambda}_4 + \hat{\lambda}_5 + \hat{\lambda}_6 + \hat{\lambda}_7 + \hat{\lambda}_8 + \hat{\lambda}_9 + \hat{\lambda}_{10} + \hat{\lambda}_{11} + \hat{\lambda}_{12}, \\
\lambda_{10,12} &= \lambda_{12,14} = \lambda_{14,15} = \hat{\lambda}_1 + \hat{\lambda}_2, \\
\lambda_{15,17} &= \lambda_{17,19} = \hat{\lambda}_3 + \hat{\lambda}_5 + \hat{\lambda}_8 + \hat{\lambda}_{10}, \\
\lambda_{15,18} &= \hat{\lambda}_1 + \hat{\lambda}_2 + \hat{\lambda}_4 + \hat{\lambda}_6 + \hat{\lambda}_7 + \hat{\lambda}_9 + \hat{\lambda}_{11} + \hat{\lambda}_{12}, \\
\lambda_{18,19} &= \hat{\lambda}_4 + \hat{\lambda}_6 + \hat{\lambda}_7 + \hat{\lambda}_9 + \hat{\lambda}_{11} + \hat{\lambda}_{12}, \\
\lambda_{18,20} &= \lambda_{20,21} = \hat{\lambda}_1 + \hat{\lambda}_2, \\
\lambda_{19,21} &= \hat{\lambda}_3 + \hat{\lambda}_4 + \hat{\lambda}_5 + \hat{\lambda}_6 + \hat{\lambda}_7 + \hat{\lambda}_8 + \hat{\lambda}_9 + \hat{\lambda}_{10} + \hat{\lambda}_{11} + \hat{\lambda}_{12} + \hat{\lambda}_{13},
\end{aligned}$$

where $\lambda_{i,j} = 0$ otherwise.

Likewise, the departure rate from the network out of node i is given by Equation (3.31)

$$d_i = \lambda_{i,0} = \sum_{k=1}^{13} \hat{\lambda}_k I_{\{k:n_{k,n_k}=i\}}.$$

This means the departure rate from node i is the sum of all external arrival rates of routes that have node i as their last node. Therefore, the departure rates are

$$\begin{aligned} d_i &= \lambda_{i,0} = 0, \qquad i = 1, 2, ..., 20, \\ d_{21} &= \lambda_{4,0} = \sum_{i=1}^{13} \hat{\lambda}_i. \end{aligned}$$

The proportion of entities at node j that go next to node i is found according to Equation (3.32) by

$$r_{ij} = \frac{\lambda_{i,j}}{\lambda_{i,0} + \sum_{h=1}^{21} \lambda_{i,h}}.$$

The values for this model are

$$\begin{aligned} r_{3,4} &= \lambda_{3,4}/(\lambda_{3,4} + \lambda_{3,5} + \lambda_{3,6} + \lambda_{3,16}), \\ r_{3,5} &= \lambda_{3,5}/(\lambda_{3,4} + \lambda_{3,5} + \lambda_{3,6} + \lambda_{3,16}), \\ r_{3,6} &= \lambda_{3,6}/(\lambda_{3,4} + \lambda_{3,5} + \lambda_{3,6} + \lambda_{3,16}), \\ r_{3,16} &= \lambda_{3,16}/(\lambda_{3,4} + \lambda_{3,5} + \lambda_{3,6} + \lambda_{3,16}), \\ r_{6,7} &= \lambda_{6,7}/(\lambda_{6,7} + \lambda_{6,8}), \\ r_{6,8} &= \lambda_{6,8}/(\lambda_{6,7} + \lambda_{6,8}), \\ r_{10,11} &= \lambda_{10,11}/(\lambda_{10,11} + \lambda_{10,12}), \\ r_{10,12} &= \lambda_{10,12}/(\lambda_{10,11} + \lambda_{10,12}), \\ r_{15,17} &= \lambda_{15,17}/(\lambda_{15,17} + \lambda_{15,18}), \\ r_{15,18} &= \lambda_{15,18}/(\lambda_{15,17} + \lambda_{15,18}), \\ r_{18,19} &= \lambda_{18,19}/(\lambda_{18,19} + \lambda_{18,20}), \\ r_{18,20} &= \lambda_{18,20}/(\lambda_{18,19} + \lambda_{18,20}), \end{aligned}$$

where $r_{i,j} = 1$ for $(i,j) \in \{(1,2), (2,3), (4,8), (5,8), (7,8), (8,9), (9,10), (11,13), (12,14), (13,15), (14,15), (16,19), (17,19), (19,21), (20,21)\}$ and $r_{i,j} = 0$ otherwise. The routing matrix R is identical in structure to that of the high resolution, single-class model but some of the nonzero entries differ depending on the $r_{i,j}$ values calculated above.

Then, by Equation (3.1), the computed arrival rates to each node are

$$\begin{aligned}
\lambda_1 &= \lambda_{0,1} = \hat{\lambda}_1 + \hat{\lambda}_3 + \hat{\lambda}_5 + \hat{\lambda}_6 + \hat{\lambda}_7 + \hat{\lambda}_8 + \hat{\lambda}_9 + \hat{\lambda}_{10} + \hat{\lambda}_{11} + \hat{\lambda}_{13}, \\
\lambda_2 &= \lambda_1 r_{1,2}, \\
\lambda_3 &= \lambda_2 r_{2,3}, \\
\lambda_4 &= \lambda_3 r_{3,4}, \\
\lambda_5 &= \lambda_3 r_{3,5}, \\
\lambda_6 &= \lambda_3 r_{3,6}, \\
\lambda_7 &= \lambda_6 r_{6,7}, \\
\lambda_8 &= \lambda_4 r_{4,8} + \lambda_5 r_{5,8} + \lambda_6 r_{6,8} + \lambda_7 r_{7,8}, \\
\lambda_9 &= \lambda_8 r_{8,9}, \\
\lambda_{10} &= \lambda_{0,10} + \lambda_9 r_{9,10} = \hat{\lambda}_2 + \hat{\lambda}_4 + \hat{\lambda}_{12} + \lambda_9 r_{9,10}, \\
\lambda_{11} &= \lambda_{10} r_{10,11}, \\
\lambda_{12} &= \lambda_{10} r_{10,12}, \\
\lambda_{13} &= \lambda_{11} r_{11,13}, \\
\lambda_{14} &= \lambda_{12} r_{12,14}, \\
\lambda_{15} &= \lambda_{13} r_{13,15} + \lambda_{14} r_{14,15}, \\
\lambda_{16} &= \lambda_3 r_{3,16}, \\
\lambda_{17} &= \lambda_{15} r_{15,17}, \\
\lambda_{18} &= \lambda_{15} r_{15,18}, \\
\lambda_{19} &= \lambda_{16} r_{16,19} + \lambda_{17} r_{17,19} + \lambda_{18} r_{18,19}, \\
\lambda_{20} &= \lambda_{18} r_{18,20}, \\
\lambda_{21} &= \lambda_{19} r_{19,21} + \lambda_{20} r_{20,21},
\end{aligned}$$

The proportion of arrivals to node j coming from node i are found using Equation (3.5)

$$\begin{aligned} p_{0,1} &= \lambda_{0,1}/\lambda_1 = \lambda_{0,1}/\lambda_{0,1} = 1, \\ p_{i,j} &= \lambda_{i,j}/\lambda_j, \end{aligned}$$

for $(i,j) \in \{(1,2), (2,3), (3,4), (3,5), (3,6), (3,16), (4,8), (5,8), (6,7), (6,8), (7,8), (8,9), (9,10), (10,11), (10,12), (11,13), (12,14), (13,15), (14,15), (15,17), (15,18), (16,19), (17,19), (18,19), (18,20), (19,21), (20,21)\}$, and $p_{i,j} = 0$ otherwise.

Next, aggregate service-time parameters are found. The aggregate service rates at each node are found by averaging the route specific service rates according to Equation (3.33)

$$\mu_j = \frac{\sum_{k=1}^{13} \sum_{l=1}^{n_k} \hat{\lambda}_k I_{\{(k,l):n_{k,l}=j\}}}{\sum_{k=1}^{13} \sum_{l=1}^{n_k} \left(\hat{\lambda}_k / \hat{\mu}_{k,l}\right) I_{\{(k,l):n_{k,l}=j\}}}, \qquad j = 1, 2, ..., 21.$$

For each node j, the numerator sums the external arrival rates for the routes that include j somewhere in their route. If a node is visited m times on a given route then the external arrival rate for that route is included m times in the sum. The denominator is equivalent to the numerator with the exception that each summand is divided by the service rate at node j according to where j occurs on the corresponding route. Define $\mathcal{R}$ as the set of all possible routes through the network. The service rates for this model configuration are thus

$$\mu_j = \frac{\lambda_j}{\sum_{i \in \mathcal{R}_j} \frac{\hat{\lambda}_j}{\hat{\mu}_{j,i}}} \qquad j = 1, 2, ..., 21,$$

where $\mathcal{R}_j$=$\{r \in \mathcal{R}$: r passes through node $j\}$. The $\mathcal{R}_j$ for this model are given in Table 3.3

Table 3.3 Routes passing through each network node.

Node (j)	$\mathcal{R}_j$
1,2,3	{1,3,5,6,7,8,9,10,11,13}
4	{1}
5	{3,7}
6	{5,6,8,9,10,11}
7	{6,10,11}
8,9	{1,3,5,6,7,8,9,10,11}
10,15	{1,2,3,4,5,6,7,8,9,10,11,12}
11,13	{3,4,5,6,7,8,9,10,11,12}
12,14,20	{1,2}
16	{13}
17	{3,5,8,10}
18	{1,2,4,6,7,9,11,12}
19	{3,4,5,6,7,8,9,10,11,12,13}
21	{1,2,3,4,5,6,7,8,9,10,11,12,13}

The traffic intensities ρ_j can now be found for each node by

$$\rho_j = \lambda_j/\mu_j, \qquad j = 1, 2, ..., 21.$$

The aggregate service squared coefficients of variation c_{sj}^2 are found using Equation (3.34)

$$c_{sj}^2 = \left\{ \frac{\sum_{k=1}^{13} \sum_{l=1}^{n_k} \left(\hat{\lambda}_k/\hat{\mu}_{k,l}^2\right) \left(\hat{c}_{sk,l}^2 + 1\right) I_{\{(k,l):n_{k,l}=j\}}}{\sum_{k=1}^{13} \sum_{l=1}^{n_k} \hat{\lambda}_k I_{\{(k,l):n_{k,l}=j\}}} \right\} - 1, \qquad j = 1, 2, ...21.$$

Hence, the aggregate service squared coefficients of variation for this model configuration are

$$c_{sj}^2 = \left\{ \frac{\mu_j^2}{\lambda_j} \left(\sum_{i \in \mathcal{R}_j} \frac{\hat{\lambda}_i}{\hat{\mu}_{j,i}^2} \left(\hat{c}_{j,i}^2 + 1\right) \right) \right\} - 1, \qquad j = 1, 2, ..., 21,$$

where $\mathcal{R}_j$ is defined as in Table 3.3.

The aggregate external arrival square coefficients of variation are found using Equation (3.35)

$$c_{0j}^2 = (1 - \hat{w}_j) + \hat{w}_j \left\{ \sum_{k=1}^{13} \hat{c}_{0k}^2 \left(\frac{\hat{\lambda}_k I_{\{k:n_{k,1}=j\}}}{\sum_{l=1}^{13} \hat{\lambda}_l I_{\{l:n_{l,1}=j\}}} \right) \right\}, \qquad j = 1, 2, ..., 21$$

where the $\hat{w}_j$ are defined by Equation (3.36)

$$\hat{w}_j = \left\{ 1 + 4\left(1 - \rho_j\right)^2 \left(\hat{v}_j - 1\right) \right\}^{-1},$$

and the $\hat{v}_j$ are defined by Equation (3.37)

$$\hat{v}_j = \left\{ \sum_{k=1}^{13} \left(\frac{\hat{\lambda}_k I_{\{k:n_{k,1}=j\}}}{\sum_{l=1}^{13} \hat{\lambda}_l I_{\{l:n_{l,1}=j\}}} \right)^2 \right\}^{-1}, \qquad j = 1, 2, ..., 21.$$

Thus, for this model configuration, the aggregate external arrival squared coefficients of variation are

$$\begin{aligned} c_{01}^2 &= 1 - \hat{w}_1 + \frac{\hat{w}_1}{\lambda_{0,1}} \sum_I c_i^2 \hat{\lambda}_i, \qquad I = \{1, 3, 5, 6, 7, 8, 9, 10, 11, 13\}, \\ c_{010}^2 &= 1 - \hat{w}_{10} + \frac{\hat{w}_{10}}{\lambda_{0,10}} \left\{ c_2^2 \hat{\lambda}_2 + c_4^2 \hat{\lambda}_4 + c_{12}^2 \hat{\lambda}_{12} \right\}, \end{aligned}$$

and $c_{0j} = 0$ otherwise where the $\hat{w}_j$ can be found using Equation (3.36) and the nonzero $\hat{v}_j$ are

$$\begin{aligned} \hat{v}_1 &= \frac{\lambda_{0,1}^2}{\hat{\lambda}_1^2 + \hat{\lambda}_3^2 + \hat{\lambda}_5^2 + \hat{\lambda}_6^2 + \hat{\lambda}_7^2 + \hat{\lambda}_8^2 + \hat{\lambda}_9^2 + \hat{\lambda}_{10}^2 + \hat{\lambda}_{11}^2 + \hat{\lambda}_{13}^2}, \\ \hat{v}_{10} &= \frac{\lambda_{0,10}^2}{\hat{\lambda}_2^2 + \hat{\lambda}_4^2 + \hat{\lambda}_{12}^2}. \end{aligned}$$

Now that the c_{0j} have been found, the remaining equations can be worked out. These remaining equations are identical to those of the high resolution, single-class configuration starting with the equations to approximate the squared coefficients of variation for the arrival processes at each node (Equations (3.7) - (3.25)).

In this chapter, the JRSOI process has been defined and queueing networks have been proposed as a viable approach for modeling it. Relevant queueing network approximation equations have been provided and four distinct model configurations have been presented based on four different class (single versus multiple) and resolution (high versus low) combinations. In Chapter 4, the models are compared to Monte-Carlo simulated values to validate their ability to produce computationally expedient numerical results.

4. Numerical Results

In this chapter, a numerical analysis of the models presented in Chapter 3 is provided. A queueing network analysis tool was developed for the purpose of obtaining performance metrics for the JRSOI process. Corresponding computer simulations were developed to be used in the model verification process. Several example problems are provided as a basis for the output comparisons.

The analytical tool was developed using MATLAB® computing software by Mathworks, Inc. It is based on the QNA approximation algorithms presented in Chapter 3. This tool requires four inputs for each stage of the network: the mean and squared coefficient of variation of its external interarrival- and service-time distributions. The number of nodes, the number of entity classes, and the routing matrix were hardcoded for each of the four defined models. In practice, however, this information is required as input. The tool outputs traffic rates, traffic variability, and congestion measures for each node and for the network as a whole.

The computer simulations were developed using Arena® simulation software. External arrival processes were modeled using *Create* nodes. The low resolution, single-class model had four *Create* nodes, one for each of the four possible entry points. Similarly, the high resolution, single-class model had two *Create* nodes, one for each of the two possible entry points. The low resolution, multi-class model, had 16 *Create* nodes based on its 16 distinct class/route combinations. Similarly, the high resolution, multi-class model, had 13 *Create* nodes based on its 13 distinct route possibilities for pax, outsize-cargo, oversize-cargo, and bulk-cargo entity classes and combat units arriving intact (complete units). No limit was placed on the total number of arrivals allowed. A single entity per arrival was specified and the first creation time for all entities was set at zero. At a given *Create* node, arrivals were governed by probability distributions for external interarrival times.

Stages were modeled using *Process* nodes. The models had four main *Process* nodes representing the four phases of JRSOI. For the high resolution models, each of these *Process* nodes was based on a submodel corresponding to the queueing network within that node's respective JRSOI phase. All *Process* nodes in the model used the same framework. They operated according to *Seize-Delay-Release* action, *Medium* priority, and *Value Added* allocation (this feature had no bearing on the model). There was a single server at each node that allowed for varying service rates depending on entity type. Delays were governed by probability distributions for service times.

One long replication of one million 24-hour days was used in all models to approximate steady-state results. A 10,000 day warm-up period was used to overcome the effect of initial conditions. The relevant statistics were initialized upon completion of the warm-up period. Each arrival and service process in the network utilized a unique random number generating stream.

A set of input parameters was prescribed for each of the four models. The data were chosen arbitrarily within sensible bounds. First, they were chosen such that the stability condition was satisfied at every node, meaning the arrival rate was less than the service rate. If the stability condition is not satisfied, the QNA equations are invalid and the simulations experience an overload of queued entities. Second, the selected traffic intensities were moderately high (between 80% and 100%) to more accurately model the high intensity activity associated with real-world operations.

The prescribed input parameters were entered in the form of parametric probability distributions. The exponential distribution was set as the default distribution for every node. While not necessary, only one non-exponential distribution was used for each test. For example, when the gamma distribution with $\alpha = 2$ was chosen to model service rates, it was used to model service rates for every node in the network (in this case, the β parameters were calculated based on the established service rates). This was done to observe the effect of the selected parameterized distributions and their associated squared coefficients of variation on model performance. In practice, any distribution can be specified for any process. Next, the input parameters used for testing are prescribed.

4.1 Model Comparisons

Each of the four models presented in Chapter 3 was tested using a prescribed set of input with the exception that the specified probability distributions, and their corresponding squared coefficients of variation, were varied from test to test. Tests are distinguished, in this analysis, by the parametric probability distributions used in the simulation models. Results are tabulated for numerical comparison.

The performance measures L_q, L, W_q, W, and throughput d for each node and for the network are directly related to each other (see Equations (3.13)-(3.17)) so testing based on these measures would produce similar or identical results. Therefore, it is reasonable to test only one of these measures as being representative of the others. The arbitrarily selected performance measure chosen for testing was L_q, the average long-run number of entities in the queue.

The *overall* category in the summary tables is based on summing the individual results across the network, and thus, represents the average long-run time spent in the network for entities arriving through node one and proceeding through every node in the network. For the high resolution models, such entities are imaginary because none of the defined entities traveled through every node. The purpose of this category is to provide an overall view of network performance. If overall network performance measures are desired for entities not fitting this description, they can be obtained by summing the performance measures for each node along the routes of those entities.

The tables include 95% confidence interval half-widths for the simulated output. It is important to note that, as the complexity of the model increases, the amount of simulation run length required to achieve sufficiently small half-widths increases. However, during this analysis, fixed run lengths were used to limit experimentation time. Hence, the accuracy of the simulations is expected to decrease for the more complex models. The run times for the simulations were primarily a function of overall throughput but also of network size and complexity.

4.1.1 Low Resolution, Single-Class Model

This model represents the flow of a single type of entity (i.e., only pax or only cargo) through the network shown in Figure 3.1. The external arrival and service rates for this model, in number of occurrences per hour, are provided in Table 4.1.

Table 4.1 Rates for low resolution, single-class model (hr^{-1}).

Node	External Arrival	Service
1	4	5
2	3	9
3	3	12
4	3	15

The first test was based on exponentially distributed interarrival and service times. This model was a product-form network; therefore, all four nodes could be analyzed individually as $M/M/1$ queues [4]. Hence, exact analytical results could be obtained for each node thus providing a basis for verifying both QNA and the simulation model. The results for this test are contained in Table 4.2. The relative error of the QNA and simulation results to the analytical results is shown. The exponential distribution has a squared coefficient of 1.

Table 4.2 Low resolution, single-class results for steady-state queue lengths (L_q) using exponential interarrival and service times.

Node	Analytical	QNA	Simulation	QNA Error (%)	Simulation Error (%)	95% CI Half-width
1	3.2000	3.2000	3.1938	0.0000	0.1938	0.0081
2	2.7222	2.7222	2.7186	0.0000	0.1322	0.0058
3	4.1667	4.1667	4.1665	0.0000	0.0048	0.0084
4	5.6333	5.6333	5.6287	0.0000	0.0817	0.0130
Overall	15.7222	15.7222	15.7076	0.0000	0.0929	-

The QNA results were exact as expected thus providing some verification of the QNA model. The low levels of relative error for the simulation output validated the simulation model and indicated that the simulated run length was sufficiently long for testing purposes. The simulation run time for this test was 5 hours and 24 minutes as compared to roughly 0.1 seconds for QNA.

For the second test, the exponential service time distributions were replaced with gamma(2, β) service time distributions (See Appendix A). The first node could be analyzed individually as an $M/G/1$ queue, but the others could not due to network dependencies. Therefore, exact analytical results could be calculated for node 1 only. The results for this test are contained in Table 4.3. The relative error of the QNA results to the simulation results is shown. The gamma(2, β) distribution is known as the Erlang distribution and has a squared coefficient of variation of 0.5.

Table 4.3 Low resolution, single-class results for steady-state queue lengths (L_q) using exponential interarrival and gamma(2, β) service times.

Node	QNA	Simulation	Relative Error (%)	95% CI Half-width
1	2.4000	2.3953	0.1962	0.0048
2	1.8263	1.7740	2.9481	0.0031
3	2.6194	2.6692	1.8657	0.0046
4	3.3234	3.5140	5.4240	0.0079
Overall	10.2071	10.3525	1.4045	-

While not shown in the table, the QNA result is exact for the $M/G/1$ Reception node. The errors for nodes 2 and 4 were slightly larger than those for nodes 1 and 3.

This indicates a slight weakness by QNA to model the particular squared coefficient of variation, traffic intensity, and network construct combinations represented by these nodes. Nevertheless, all errors were generally low thus affirming QNA. The simulation run time was 5 hours and 18 minutes as compared to about 0.1 seconds for QNA.

The gamma(2, β) distributed service times were then replaced with Weibell(2, γ) distributed service times (See Appendix A). The first node could be analyzed individually as an $M/G/1$ queue but the others could not because of network dependencies. Therefore, exact analytical results could be calculated for node 1 only. The results for this test are contained in Table 4.4. The relative error of the QNA results to the simulation results is shown. The Weibell(2, γ) distribution is known as the Rayleigh distribution and has a squared coefficient of variation of $(4/\pi - 1) \approx .2732$.

Table 4.4 Low resolution, single-class results for steady-state queue lengths (L_q) using exponential interarrival and Weibell(2, γ) service times.

Node	QNA	Simulation	Relative Error (%)	95% CI Half-width
1	2.0372	2.0338	0.1672	0.0044
2	1.4160	1.3345	6.1072	0.0020
3	1.9078	1.9686	3.0885	0.0037
4	2.2581	2.5238	10.5278	0.0044
Overall	7.6752	7.8607	2.3598	-

The QNA result was exact for the $M/G/1$ Reception node as compared to analytical results. The errors associated with nodes 2 and 4 were slightly large indicating a possible cause for concern. The explanation for this behavior is the same as for the previous test. The table reveals a degradation in performance for this test as compared to the previous one. It is suspected that this was due to the squared coefficient of variation for the service-time distribution being further away from 1. The simulation run time was 5 hours and 45 minutes as compared to about 0.1 seconds for QNA.

Next, the gamma(2, β) distribution was used for all interarrival and service times. All four nodes were $G/G/1$ queues. Therefore, exact analytical results could not be calculated for nodes 2 through 4. Through great expenditure of effort, exact analytical results could have been calculated for node 1; however, this was not done here. The results for this test are contained in Table 4.5. The relative error of the QNA results to the simulation results is shown.

Table 4.5 Low resolution, single-class results for steady-state queue lengths (L_q) using gamma(2, β) interarrival and service times.

Node	QNA	Simulation	Relative Error (%)	95% CI Half-width
1	1.5347	1.5095	1.6694	0.0053
2	1.4245	1.3185	8.0394	0.0034
3	2.1497	2.0021	7.3723	0.0056
4	2.8438	2.7647	2.8611	0.0062
Overall	7.9527	7.5948	4.7124	-

The complexity of the model increased greatly, due to the non-exponential interarrival times, yet the performance of QNA was only slightly worse. All relative errors were low thus affirming QNA. The simulation run time was 5 hours and 33 minutes as compared to 0.1 seconds for QNA.

Finally, the gamma(2, β) distribution was replaced with the Weibull(2, γ) distribution in all instances. All four nodes were $G/G/1$ queues. Therefore, exact analytical results could not be calculated for nodes 2 through 4. Although possible, exact analytical results were not calculated for node 1. The results for this test are contained in Table 4.6.

Table 4.6 Low resolution, single-class results for steady-state queue lengths (L_q) using Weibull(2, γ) interarrival and service times.

Node	QNA	Simulation	Relative Error (%)	95% CI Half-width
1	0.7443	0.7623	2.3613	0.0011
2	0.8098	0.7280	11.2363	0.0007
3	1.2031	1.1291	6.5539	0.0010
4	1.5412	1.5180	1.5283	0.0018
Overall	4.2984	4.1374	3.8913	-

For the last two models, the larger errors corresponded to nodes 2 and 3 thus indicating the contribution of non-exponential interarrival times to this network. It can not be concluded, for these models, that the one with one with the squared coefficient closer to 1 produced more accurate results. All the errors were reasonably low thus affirming QNA. The simulation run time was 5 hours and 55 minutes as compared to about 0.1 seconds for QNA.

4.1.2 Low Resolution, Multi-Class Model

The external arrival and service rates for this model were specified for each class/route. Each of the four defined entity classes, Pax, Outsize Cargo (CT), Oversize Cargo (CV) and Bulk Cargo (CB), had a class-specific service rate at each node regardless of where it entered the network. The rates, in number of occurrences per hour, are given in Table 4.7.

Table 4.7 Rates for low resolution, multi-class model (hr^{-1}).

		Service by Node			
Class/Route	External Arrival	1	2	3	4
P1	4.5	20	30	30	32
CT1	0.5	2	20	20	24
CV1	0.5	2	20	20	24
CB1	0.5	2	20	20	24
P2	2.0	-	30	30	32
CT2	4.0	-	20	20	24
CV2	4.0	-	20	20	24
CB2	4.0	-	20	20	24
P3	0.5	-	-	30	32
CT3	0.5	-	-	20	24
CV3	0.5	-	-	20	24
CB3	0.5	-	-	20	24
P4	0.5	-	-	-	32
CT4	0.5	-	-	-	24
CV4	0.5	-	-	-	24
CB4	0.5	-	-	-	24

This model was first tested using exponentially distributed interarrival and service times. Node 1 was a $G/M/1$ queue and so exact analytical results could have been obtained for it. However, this study is primarily concerned with the performance of nodes affected by network dependencies (i.e., all nodes other than node 1); therefore, the effort to calculate analytical results for node 1 was not warranted. The results for this test are contained in Table 4.8. The relative error of the QNA results to the simulation results is shown.

Table 4.8 Low resolution, multi-class results for steady-state queue lengths (L_q) using exponential interarrival and service times.

Node	QNA	Simulation	Relative Error (%)	95% CI Half-width
1	95.0840	91.8940	3.4714	1.8263
2	9.3302	9.3553	0.2683	0.0197
3	68.4020	67.2580	1.7009	0.9020
4	11.3730	11.3580	0.1321	0.0324
Overall	184.1892	179.8653	2.4040	-

The relative errors for all nodes were small thus affirming QNA. The larger errors occurred at the nodes with larger confidence interval half-widths relative to the size of the performance measure, thus indicating that the simulation contributed to these errors. Increasing the simulation run length would shrink these confidence intervals and provide a better benchmark to compare against. The simulation run time was 14 hours and 43 minutes as compared to approximately 0.5 seconds for QNA.

The exponentially distributed service times were replaced with lognormally distributed service times (denoted $\mathrm{LN}(\mu, \sigma^2)$) with shape parameter $\sigma = 3/4$ and scale parameter μ. The scale parameter was calculated based on the prescribed service rates and the distribution mean, $e^{\mu+9/32}$. The $\mathrm{LN}(\mu, 9/16)$ distribution has a squared coefficient of variation of $(e^{9/16} - 1) \approx 0.7551$. All four nodes for this model were $G/G/1$ queues. Exact analytical results could not be calculated for nodes 2 through 4 because their interarrival-time distributions were unknown. Though possible, the exact analytical results were not calculated for node 1 because the performance at node 1 was not of primary interest for the reason just mentioned. The results for this test are contained in Table 4.9. The relative error of the QNA results to the simulation results is shown.

Table 4.9 Low resolution, multi-class results for steady-state queue lengths (L_q) using exponential interarrival and $\mathrm{LN}(\mu, 9/16)$ service times.

Node	QNA	Simulation	Relative Error (%)	95% CI Half-width
1	81.3468	80.4130	1.1613	1.5422
2	8.7479	7.8809	11.0013	0.0181
3	51.5498	57.3170	10.0619	0.7532
4	8.8216	8.6317	2.2000	0.0215
Overall	150.4661	154.2426	2.4484	-

The size of the confidence interval half-widths was larger at nodes 1 and 3 relative to the size of the performance measure, thus possibly explaining the larger relative error at node 3 but not node 2. The larger errors at nodes 2 and 3 might reveal a slight weakness by QNA to model the particular squared coefficient of variation, traffic intensity, and network construct combinations represented by these nodes. Nevertheless, all errors were relatively small thus affirming QNA. The simulation run time was 14 hours and 41 minutes. The run time for QNA was about 0.5 seconds for this test and the remaining tests for this model.

Next, the LN($\mu, 9/16$) service times were replaced with Weibull(4/5, γ). The results are contained in Table 4.10. The Weibull(4/5, γ) has a squared coefficient of variation of $(8\Gamma(2.5)/5\Gamma(1.25)^2 - 1) \approx 1.5889$ where the $\Gamma(\cdot)$ is the gamma function (See Appendix A).

Table 4.10 Low resolution, multi-class results for steady-state queue lengths (L_q) using exponential interarrival and Weibull(4/5, γ) service times.

Node	QNA	Simulation	Relative Error (%)	95% CI Half-width
1	119.9952	118.1400	1.5703	2.7654
2	13.7842	12.5240	10.0623	0.0356
3	97.1904	91.3940	6.3422	1.4827
4	17.4047	17.5940	1.0759	0.0567
Overall	248.3745	239.6520	3.6397	-

This test had the same error pattern as the previous one; however, the errors for this test were slightly lower. The errors were generally small thus affirming QNA. The simulation run time was 14 hours and 15 minutes.

The model was then tested using LN($\mu, 9/16$) distributed interarrival and service times. No exact analytical results were calculated. The results for this test are contained in Table 4.11.

Table 4.11 Low resolution, multi-class results for steady-state queue lengths (L_q) using LN($\mu, 9/16$) interarrival and service times.

Node	QNA	Simulation	Relative Error (%)	95% CI Half-width
1	76.6870	68.8270	11.4199	1.5312
2	8.2435	6.7354	22.3907	0.0116
3	50.1659	50.7500	1.1509	0.4611
4	8.7159	8.3822	3.9811	0.0183
Overall	143.8123	134.6946	6.7692	-

Again, the size of the half-widths were larger for nodes 1 and 3. The half-width for node 1 in particular was over 2% of the size of the performance measure thus at least partially explaining the large relative error for node 1. The error of QNA relative to the simulation for node 2 was somewhat large. This might indicate a weakness by QNA to model the particular squared coefficient of variation, traffic intensity, and network construct combinations represented by this node. Simulation run time was 15 hours and 33 minutes.

The simulation was run a second time, using a different set of random number generating streams, to see if the value of L_q for node 2, in particular, would be closer to the QNA value. The relative error for node 2 was 22.4% for both runs. One possible reason for this higher level of error was the large size of the half-width for node 1. The performance at node 1 directly affects the performance at node 2, so it is possible that the outcome for node 1 and hence for node 2 would be more accurate given a longer run time. A more likely explanation is that the particular combination of distributions, squared coefficients of variation, traffic intensity, and network structure specified for node 2 was not well captured by the QNA equations. The simulation was run a third time using identical squared coefficients of variation but using the gamma(α, β) distribution where $\alpha = (e^{9/16} - 1)^{-1}$. The purpose was to investigate the possibility that the nature of the distribution itself led to the high error at node 2. The results for this test are contained in Table 4.12.

Table 4.12 Low resolution, multi-class results for steady-state queue lengths (L_q) using gamma(α, β) interarrival and service times where $\alpha = (e^{9/16} - 1)^{-1}$.

Node	QNA	Simulation	Relative Error (%)	95% CI Half-width
1	76.6870	69.0910	10.9942	1.0870
2	8.2435	7.2399	13.8621	0.0116
3	50.1659	51.0430	1.7184	0.5398
4	8.7159	8.6505	0.7560	0.0129
Overall	143.8123	136.0244	5.7254	-

The errors for nodes 1 and 2 were higher, just as in the previous test, thus revealing the impact the characterization of these nodes had on performance. However, it is also noted that the distribution does in fact significantly affect performance witnessed by the reduction in relative error from 22.4% to 13.9% for node 2. The simulation run time was 14 hours and 22 minutes.

The final test for this model utilized the Weibull(4/5, γ) distribution for interarrival and service times. No exact analytical results were calculated. The results for this test are contained in Table 4.13.

Table 4.13 Low resolution, multi-class results for steady-state queue lengths (L_q) using Weibull(4/5, γ) interarrival and service times.

Node	QNA	Simulation	Relative Error (%)	95% CI Half-width
1	131.1516	143.0600	8.3241	3.8391
2	14.9970	14.6120	2.6348	0.0306
3	100.5134	106.0000	5.1760	1.4291
4	17.6540	18.0860	2.3886	0.0614
Overall	264.3160	281.7580	6.1904	-

The larger errors in this test corresponded to the nodes having larger confidence interval half-widths relative to the performance measure thus suggesting the simulation contributed to these errors. Regardless, the errors of QNA relative to the simulation were generally small thus affirming QNA. The simulation run time was 14 hours and 44 minutes.

4.1.3 High Resolution, Single-Class Model

The external arrival and service rates for this model were specified for each node. These rates, in number of occurrences per hour, are given in Table 4.14.

Table 4.14 Rates for high resolution, single-class model (hr^{-1}).

Node	External Arrival	Service
1	54	55.00
2	0	56.00
3	0	55.70
4	0	23.00
5	0	12.25
6	0	18.50
7	0	1.80
8	0	51.50
9	0	49.20
10	48	99.00
11	0	74.20
12	0	32.20
13	0	70.10
14	0	29.75
15	0	97.60
16	0	6.75
17	0	39.50
18	0	61.30
19	0	88.10
20	0	19.30
21	0	105.00

The first test was based on exponentially distributed interarrival and service times. This model, through node eight, was a Jackson Network; therefore, the first eight nodes could be analyzed individually as $M/M/1$ queues and node nine (fork-join node) could be analyzed as an $M/G/1$ queue. Exact analytical results were tabulated through node nine thus providing a basis for verifying both QNA and the simulation model for this portion of the network. Results for this test are contained in Table 4.15.

Table 4.15 High resolution, single-class results for steady-state queue lengths (L_q) using exponential interarrival and service times.

Node	Analytical	QNA	Simulation	QNA Error Relative to Analytical (%)	Simulation Error Relative to Analytical (%)	QNA Error Relative to Simulation (%)	95% CI Half-width
1	53.0182	53.0182	51.7110	0.0000	2.4655	2.5279	1.7269
2	26.0357	26.0357	25.5910	0.0000	1.7081	1.7377	0.5092
3	30.7952	30.7952	30.4390	0.0000	1.1568	1.1702	0.5812
4	14.4894	14.4894	14.3480	0.0000	0.9762	0.9855	0.1799
5	6.5666	6.5666	6.5732	0.0000	0.0998	0.1004	0.0772
6	6.1678	6.1678	6.1580	0.0000	0.1589	0.1591	0.0571
7	8.1000	8.1000	7.9545	0.0000	1.7963	1.8292	0.3171
8	15.8149	15.8149	15.8040	0.0000	0.0691	0.0690	0.1690
9	58.0881	57.1003	55.7220	1.7300	4.0733	2.4735	3.1776
10	-	33.7401	34.3040	-	-	1.6438	0.6656
11	-	9.3212	9.1860	-	-	1.4718	0.0595
12	-	8.0837	8.0191	-	-	0.8056	0.0398
13	-	26.2805	25.2950	-	-	3.8960	0.2929
14	-	36.6482	36.1480	-	-	1.3838	0.9195
15	-	95.6060	88.7920	-	-	7.6741	3.6263
16	-	3.2000	3.1979	-	-	0.0657	0.0378
17	-	43.9520	42.6040	-	-	3.1640	1.1649
18	-	16.4078	16.3410	-	-	0.4088	0.1604
19		23.2976	23.0850	-	-	0.9209	0.2485
20	-	8.1932	8.2091	-	-	0.1937	0.0639
21	-	33.0286	32.7650	-	-	0.8045	0.4322
Overall	-	555.8470	542.2468	-	-	2.5081	-

The QNA results were exact as expected for the first eight nodes thus providing some verification of the QNA model. The error for node nine indicated that QNA did a better job modeling the performance of the fork-join construct than did the simulation. The largest errors corresponded invariably to nodes whose confidence interval half-widths were relatively large, ranging from 2% to 6% of the size of the performance measure. This indicated that the selected simulation run lengths might not be sufficiently long enough for testing purposes and hence the simulation results might contribute significantly to the errors for this model's tests. The 7.7% error for node 15 was noticeably higher than for the others. This is caused by QNA's inability to perfectly model the activity unique to this node. The errors for the remaining nodes were all below 5%. The simulation run time was 7 hours and 11 minutes. The QNA run times for all the tests related to this model were all approximately 0.5 seconds.

Next, the exponential distribution was replaced with the LN$(\mu, 9/16)$ distribution for service times. The first node could be analyzed individually as an $M/G/1$ queue, but the others could not because of network dependencies. The results for this test are contained in Table 4.16. The relative error of the QNA results to the simulation results is shown.

Table 4.16 High resolution, single-class results for steady-state queue lengths (L_q) using exponential interarrival and LN($\mu, 9/16$) service times.

Node	QNA	Simulation	Relative Error (%)	95% CI Half-width
1	46.5249	45.4390	2.3898	1.4335
2	19.7554	20.0210	1.3266	0.3668
3	23.2423	24.0170	3.2256	0.4491
4	12.0022	11.9730	0.2439	0.1846
5	5.6009	5.6674	1.1734	0.0532
6	5.1843	5.1868	0.0482	0.0344
7	7.0248	7.0996	1.0536	0.2746
8	14.9480	15.1920	1.6061	0.1993
9	47.4305	51.1630	7.2953	2.5109
10	28.8719	29.4360	1.9164	0.5582
11	7.4003	7.3384	0.8435	0.0517
12	6.8065	6.8115	0.0734	0.0407
13	20.0026	20.6520	3.1445	0.2863
14	28.2626	31.1390	9.2373	0.9307
15	72.2272	71.9970	0.3197	2.9283
16	2.7687	2.7529	0.5739	0.0339
17	36.4130	36.2260	0.5162	1.0034
18	13.1859	13.1930	0.0538	0.1420
19	18.1404	18.2290	0.4860	0.1825
20	6.9000	6.9518	0.7451	0.0665
21	25.0692	25.5190	1.7626	0.3469
Overall	447.7616	456.0044	1.8076	-

While not shown in the table, the QNA result was exact for the $M/G/1$ Reception node. Again, the larger errors corresponded to nodes with relatively large 95% CI half-widths. The volatility of the fork-join construct may have contributed to the error at node 9. The error at node 14 is attributed to QNA's inability to rightly model the activity there. Even so, all errors were relatively low thus affirming QNA. The simulation run time was 7 hours and 56 minutes.

LN($\mu, 9/16$) was then replaced with Weibull(4/5, γ) as the service-time distribution. Node 1 could be analyzed individually as an $M/G/1$ queue, but the others could not. The results for this test are contained in Table 4.17. The relative error of the QNA results to the simulation results is shown. The Weibull(4/5, γ) distribution has a squared coefficient of variation of $(8\Gamma(2.5)/5\Gamma(1.25)^2 - 1) \approx 1.5889$.

Table 4.17 High resolution, single-class results for steady-state queue lengths (L_q) using exponential interarrival and Weibull(4/5, γ) service times.

Node	QNA	Simulation	Relative Error (%)	95% CI Half-width
1	68.6292	66.6380	2.9881	2.1843
2	41.0917	38.6480	6.3230	0.8649
3	48.9074	46.3000	5.6315	0.9498
4	20.4621	20.1440	1.5791	0.3114
5	8.8868	8.8853	0.0169	0.0858
6	8.5286	8.5187	0.1162	0.0879
7	10.6846	10.3670	3.0636	0.5287
8	24.3295	24.1110	0.9062	0.3095
9	80.2182	69.2470	15.8436	2.6266
10	45.4436	46.3570	1.9704	1.0410
11	13.9178	13.6120	2.2465	0.0851
12	11.1503	10.9820	1.5325	0.0837
13	41.3320	37.0920	11.4310	0.5546
14	56.7735	50.7200	11.9351	1.3167
15	151.7657	127.4600	19.0693	5.2391
16	4.2364	4.2404	0.0943	0.0494
17	62.0696	58.8700	5.4350	1.6288
18	24.1374	23.9920	0.6060	0.3529
19	35.6666	34.9270	2.1176	0.4248
20	11.2987	11.1570	1.2701	0.1510
21	52.1210	50.2070	3.8122	0.6759
Overall	821.6507	762.4754	7.7609	-

The QNA result was exact for the $M/G/1$ Reception node as compared to analytical results. The largest errors occurred for nodes with the largest confidence interval half-widths relative to their means, indicating the simulation results contributed to these errors. The errors associated with nodes 9 and 15 were relatively large. Error at node nine was not unexpected due to the high volatility of the fork-join construct. The error at node 15 is attributed to QNA. The errors for the remaining 19 nodes were under 12%. Errors were generally larger for this test than for the previous one. The simulation run time was 7 hours and 29 minutes.

For the next test, all interarrival and service times were distributed LN(μ, 9/16). All twenty-one nodes are G/G/1 queues. Hence, exact analytical results could only be calculated for node 1 because the interarrival-time distributions for the remaining nodes were unknown. Exact analytical results were not calculated for node 1 because it would have contributed little to the analysis. The results for this test are contained in Table 4.18.

Table 4.18 High resolution, single-class results for steady-state queue lengths (L_q) using LN($\mu, 9/16$) interarrival and service times.

Node	QNA	Simulation	Relative Error (%)	95% CI Half-width
1	40.0120	38.8400	3.0175	1.1020
2	19.6391	19.3050	1.7306	0.3325
3	23.2327	23.2580	0.1088	0.3926
4	12.0021	11.9240	0.6550	0.2096
5	5.6009	5.6363	0.6281	0.0651
6	5.1842	5.1388	0.8835	0.0502
7	7.0248	6.9410	1.2073	0.2129
8	12.2604	12.2460	0.1176	0.1323
9	47.4656	47.9240	0.9565	1.7195
10	26.4605	26.6530	0.7222	0.4495
11	7.3808	7.2397	1.9490	0.0402
12	6.7993	6.7212	1.1620	0.0346
13	19.9934	20.1510	0.7821	0.2827
14	28.2565	29.5030	4.2250	0.8861
15	72.2253	68.6070	5.2740	2.1506
16	2.7687	2.7568	0.4317	0.0364
17	36.4130	35.8230	1.6470	0.9669
18	13.1859	13.1040	0.6250	0.1421
19	18.1404	18.1040	0.2011	0.2126
20	6.9000	6.9009	0.0130	0.0671
21	25.0692	25.4340	1.4343	0.3364
Overall	436.0148	432.2107	0.8801	-

This model was the most complex tested so far, yet the errors of QNA relative to the simulation were very small being at or below 5%. This indicates that there is not a direct relationship between network complexity and QNA performance. The largest errors occurred at nodes 14 and 15. The simulation run time was 7 hours and 42 minutes.

Finally, the model was tested using the Weibull(4/5, β) throughout. Again, all nodes are $G/G/1$ queues. Results for this test are contained in Table 4.19.

Table 4.19 High resolution, single-class results for steady-state queue lengths (L_q) using Weibull(4/5, β) interarrival and service times.

Node	QNA	Simulation	Relative Error (%)	95% CI Half-width
1	84.2402	81.5160	3.3419	2.9717
2	41.3680	40.7560	1.5016	0.9699
3	48.9303	48.3400	1.2211	1.1641
4	20.4623	20.1830	1.3838	0.3778
5	8.8869	8.9568	0.7804	0.1344
6	8.5287	8.5544	0.3004	0.0711
7	10.6846	10.2300	4.4438	0.3261
8	24.3295	24.4670	0.5620	0.3881
9	80.2182	76.6640	4.6361	4.1881
10	51.2006	52.5000	2.4750	1.2586
11	13.9636	13.8540	0.7911	0.1091
12	11.1673	11.1350	0.2901	0.0996
13	41.3538	38.2890	8.0044	0.5687
14	56.7881	51.4460	10.3839	1.7450
15	151.7702	136.9000	10.8621	6.9603
16	4.2365	4.2615	0.5866	0.0504
17	62.0696	59.2230	4.8066	2.1444
18	24.1374	24.1280	0.0390	0.3059
19	35.6666	35.2510	1.1790	0.4943
20	11.2987	11.2510	0.4240	0.1555
21	52.1210	51.3050	1.5905	0.8127
Overall	843.4221	809.2107	4.2277	-

This test revealed a similar error pattern to that of the previous test. All errors were reasonable small thus affirming QNA. The simulation run time was 8 hours and 9 minutes.

4.1.4 High Resolution, Multi-Class Model

The external arrival and service rates for this model were specified for each route. Each of the 13 defined routes had a route-specific service rate at each node along its path. The rates, in number of occurrences per hour, are given in Tables 4.20 and 4.21. The service rates are rounded to the nearest tenth.

Table 4.20 Rates for high resolution, multi-class model (hr^{-1}).

		Service by Node									
Route	External Arrival	1	2	3	4	5	6	7	8	9	10
1	16	64.8	68.4	67.5	18.0	-	-	-	66.6	62.1	124.2
2	4	57.6	60.8	60.0	-	9.6	-	-	59.2	55.2	110.4
3	10	57.6	60.8	60.0	-	-	38.4	-	59.2	55.2	110.4
4	2	36.0	38.0	37.5	-	-	24.0	4.5	37.0	34.5	69.0
5	4	57.6	60.8	60.0	-	9.6	-	-	59.2	55.2	110.4
6	7	50.4	53.2	52.5	-	-	33.6	-	51.8	48.3	96.6
7	7	50.4	53.2	52.5	-	-	33.6	-	51.8	48.3	96.6
8	1	36.0	38.0	37.5	-	-	24.0	4.5	37.0	34.5	69.0
9	1	36.0	38.0	37.5	-	-	24.0	4.5	37.0	34.5	69.0
10	2	72.0	76.0	75.0	-	-	-	-	-	-	138.0
11	16	64.8	68.4	67.5	-	-	-	-	-	-	124.2
12	2	72.0	76.0	75.0	-	-	-	-	-	-	138.0
13	30	50.4	53.2	52.5	-	-	-	-	-	-	96.6

Table 4.21 Rates for high resolution, multi-class model (hr^{-1}).

	Service by Node										
Route	11	12	13	14	15	16	17	18	19	20	21
1	-	20.3	-	18.9	119.7	-	-	105.8	-	21.6	126.0
2	100.0	-	95.2	-	106.4	-	24.8	-	96.0	-	112.0
3	100.0	-	95.2	-	106.4	-	24.8	-	96.0	-	112.0
4	62.5	-	59.5	-	66.5	-	-	58.8	60.0	-	70.0
5	100.0	-	95.2	-	106.4	-	-	94.0	96.0	-	112.0
6	87.5	-	83.3	-	93.1	-	21.7	-	84.0	-	98.0
7	87.5	-	83.3	-	93.1	-	-	82.3	84.0	-	98.0
8	62.5	-	59.5	-	66.5	-	15.5	-	60.0	-	70.0
9	62.5	-	59.5	-	66.5	-	-	58.8	60.0	-	70.0
10	125.0	-	119.0	-	-	2.7	-	-	120.0	-	140.0
11	112.5	-	107.1	-	119.7	-	-	105.8	108.0	-	126.0
12	-	22.5	-	21.0	133.0	-	-	117.5	-	24.0	140.0
13	87.5	-	83.3	-	93.1	-	-	82.3	84.0	-	98.0

The model is first tested using exponentially distributed interarrival and service times. Exact analytical results could not be obtained for any node under the current set of assumptions. (Exact steady-state results could be obtained by changing queuing disciplines from FCFS to LCFS-PR or by using a single service process at each node independent of routes [4]). Therefore, computer simulation was the only available basis for comparison. The results for this test are contained in Table 4.22. The relative error of the QNA results to the simulation results is shown. The simulation run time was 7 hours and 50 minutes. The QNA run times for this test and the other tests for this model were about 0.5 seconds.

Table 4.22 High resolution, multi-class results for steady-state queue lengths (L_q) using exponential interarrival and service times.

Node	QNA	Simulation	Relative Error (%)	95% CI Half-width
1	41.1312	40.7390	0.9627	0.9290
2	12.0811	11.7360	2.9405	0.0839
3	14.8531	14.4370	2.8822	0.1071
4	7.1799	7.4688	3.8681	0.0776
5	4.1868	4.2243	0.8877	0.0459
6	4.7590	4.4041	8.0584	0.0273
7	7.1360	6.7497	5.7232	0.1003
8	11.4645	11.5820	1.0145	0.0891
9	70.7920	67.5830	4.7482	2.4770
10	15.5430	15.3480	1.2705	0.1347
11	7.3158	6.9108	5.8604	0.0319
12	6.4016	6.5279	1.9348	0.0566
13	13.6926	12.8430	6.6153	0.0936
14	15.2721	15.2890	0.1105	0.2433
15	64.2096	59.0640	8.7119	1.6811
16	2.1190	2.1726	2.4671	0.0344
17	19.0971	18.2460	4.6646	0.3398
18	5.6809	5.5555	2.2572	0.0234
19	16.1337	15.4220	4.6148	0.1102
20	3.8854	4.0011	2.8917	0.0219
21	18.5553	18.0920	2.5608	0.1394
Overall	361.4897	348.3958	3.7583	-

The errors were higher than expected considering that all network processes were exponentially distributed thus indicating possible cause for concern. No correlation could be detected between error size and confidence interval half-width size relative to the performance measure. The errors of QNA relative to the simulation were small for all nodes and overall.

The exponential distribution was replaced with LN(μ, 9/16) for service times. Exact results could not be obtained for any node. The results for this test are contained in Table 4.23.

Table 4.23 High resolution, multi-class results for steady-state queue lengths (L_q) using exponential interarrival and LN(μ, 9/16) service times.

Node	QNA	Simulation	Relative Error (%)	95% CI Half-width
1	36.0938	35.7670	0.9137	0.77617
2	9.2665	8.9324	3.7403	0.0642
3	11.3269	11.2770	0.4425	0.0865
4	6.0418	6.4123	5.7780	0.0639
5	3.5983	3.6450	1.2812	0.0361
6	3.8844	3.5378	9.7970	0.0210
7	6.1547	5.8702	4.8465	0.0989
8	9.0483	9.2114	1.7706	0.0597
9	59.4708	63.5250	6.3821	2.2194
10	13.3230	13.1220	1.5318	0.1103
11	5.7565	5.3757	7.0837	0.0232
12	5.4884	5.3457	2.6694	0.0463
13	10.4715	10.3530	1.1446	0.0740
14	11.8776	12.6950	6.4388	0.2021
15	48.8437	48.8020	0.0854	1.2317
16	1.8437	1.8960	2.7584	0.0301
17	16.2534	15.4550	5.1660	0.2789
18	4.4583	4.2171	5.7196	0.0162
19	12.6896	12.0900	4.9595	0.0698
20	3.3060	3.0391	8.7822	0.0173
21	14.2415	14.0900	1.0752	0.1080
Overall	293.4387	294.6587	0.4140	-

The errors were slightly larger than for those of the previous test. No correlations were found to explain why some errors were larger than others. All errors were reasonably small. The simulation run time was 7 hours and 50 minutes.

Next, the service-time distributions were changed to Weibull(4/5, γ). The results for this test are contained in Table 4.24.

Table 4.24 High resolution, multi-class results for steady-state queue lengths (L_q) using exponential interarrival and Weibull(4/5, γ) service times.

Node	QNA	Simulation	Relative Error (%)	95% CI Half-width
1	53.2422	52.8080	0.8222	1.4284
2	18.8243	18.0710	4.1686	0.1599
3	23.3046	22.3620	4.2152	0.1738
4	9.9138	10.1480	2.3078	0.1228
5	5.6012	5.6430	0.7407	0.0666
6	6.8554	6.5385	4.8467	0.0459
7	9.4948	8.9387	6.2213	0.1584
8	17.2536	17.5730	1.8176	0.1408
9	97.9828	80.4490	21.7949	2.9282
10	20.8803	20.7900	0.4343	0.2046
11	11.0435	10.5040	5.1361	0.0560
12	8.5961	9.4256	8.8004	0.0874
13	21.4063	19.4270	10.1884	0.1436
14	23.4041	22.0290	6.2422	0.4231
15	101.1158	83.9840	20.3989	2.5639
16	2.7659	2.8341	2.4064	0.0487
17	25.9321	24.9970	3.7408	0.4966
18	8.6028	8.5964	0.0744	0.0401
19	24.3947	23.7130	2.8748	0.1812
20	5.2769	6.3457	16.8429	0.0397
21	28.8969	28.3520	1.9219	0.2632
Overall	524.7881	483.5290	8.5329	-

The errors for nodes 9, 15, and 20 were somewhat large. The error for node 9 can be partially explained by the difficulty in modeling the fork-join construct due to its volatile nature. The errors at nodes 9 and 15 did correspond to largest confidence interval half-width sizes relative to the performance measures. These two nodes were the same two nodes that experienced larger errors in the single-class version of this test thus implicating the service-time squared coefficients of variation or the Weibull distribution. No causes for the the larger error size at node 20 were detected. It is expected that these errors would decrease if the simulation run lengths were increased. The simulation run time was 7 hours and 36 minutes.

Next, LN(μ, 9/16) was used for all interarrival and service times. All nodes were $G/G/1$ queues. No exact analytical results could be calculated. The results for this test are contained in Table 4.25.

Table 4.25 High resolution, multi-class results for steady-state queue lengths (L_q) using LN(μ, 9/16) interarrival and service times.

Node	QNA	Simulation	Relative Error (%)	95% CI Half-width
1	31.2504	30.7960	1.4755	0.64856
2	9.2002	8.7384	5.2847	0.0704
3	11.3152	11.0260	2.6229	0.0797
4	6.0416	5.5251	9.3482	0.0527
5	3.5982	3.0126	19.4384	0.0257
6	3.8842	3.2693	18.8083	0.0178
7	6.1547	4.8917	25.8192	0.0798
8	9.0482	8.8461	2.2846	0.0451
9	59.4707	56.5460	5.1722	1.7474
10	12.2901	11.8680	3.5566	0.0814
11	5.7218	5.1972	10.0939	0.0209
12	5.4817	4.9293	11.2065	0.0375
13	10.4581	9.9503	5.1034	0.0623
14	11.8739	11.6890	1.5818	0.1631
15	48.8370	45.3640	7.6559	1.0024
16	1.8498	1.4683	25.9824	0.0219
17	16.2534	13.7860	17.8979	0.2788
18	4.4583	4.0931	8.9223	0.0145
19	12.6896	11.7830	7.6941	0.0638
20	3.3060	2.9830	10.8280	0.0179
21	14.2415	13.8710	2.6710	0.0952
Overall	287.4246	269.6334	6.5983	-

The largest errors did not directly correspond to the largest relative half-width sizes. The relative errors for five of the nodes were noticeably large. The maximum error was about 26%. It is expected that a longer simulation run length would only marginally decrease the error size. The larger errors here were likely caused by inability of QNA to capture all the activity occurring at these nodes. The simulation run time was 11 hours and 10 minutes.

Finally, all interarrival and service times were distributed Weibull(4/5, β). All nodes were $G/G/1$ queues. No exact results could be obtained. The results for this test are contained in Table 4.26.

Table 4.26 High resolution, multi-class results for steady-state queue lengths (L_q) using Weibull(4/5, β) interarrival and service times.

Node	QNA	Simulation	Relative Error (%)	95% CI Half-width
1	64.8406	64.5480	0.4533	1.7843
2	18.9806	18.7510	1.2245	0.1787
3	23.3324	23.0650	1.1593	0.1940
4	9.9143	11.9420	16.9796	0.1711
5	5.6014	6.7318	16.7919	0.0894
6	6.8559	7.0876	3.2691	0.0426
7	9.4948	10.7300	11.5116	0.2343
8	17.2538	18.3600	6.0251	0.1329
9	97.9831	97.1570	0.8503	4.4447
10	23.3279	23.3010	0.1154	0.2024
11	11.1245	11.0030	1.1042	0.0540
12	8.6120	10.3490	16.7842	0.0896
13	21.4377	20.4580	4.7888	0.1834
14	23.4127	24.1260	2.9566	0.4224
15	101.1319	91.9550	9.9798	3.2186
16	2.7660	3.5634	22.3775	0.0719
17	25.9321	28.6830	9.5907	0.8103
18	8.6029	8.8637	2.9423	0.0368
19	24.3947	24.4360	0.1690	0.2426
20	5.2769	6.4738	18.4884	0.0458
21	28.8969	28.9260	0.1006	0.2812
Overall	539.1731	540.5103	0.2474	-

The errors of QNA relative to the simulation were small for all but five of the nodes. All errors were below 23%. It is expected that a longer simulation run length would only marginally decrease the error size. The larger errors here were likely caused by inability of QNA to adequately capture the activity occurring at these nodes. The overall error was very small. The simulation run time was 7 hours and 29 minutes as compared to about 0.5 seconds for QNA.

Traffic intensities had no discernable impact on performance for any test for any model. However, some level of correlation was detected between traffic intensity and the size of simulation confidence interval half-widths relative to their means. In particular, the half-widths that were considerably larger than average corresponded to the traffic intensities that were extremely high, ranging between .97 and 1.

4.2 Summary

The results of this chapter affirm the ability of QNA to provide highly accurate steady-state approximations for key performance measures. It was demonstrated that QNA was just as effective for the higher resolution models as for the lower. Furthermore, QNA generally performed as well for the multi-class models as for the single-class models. Performance did degrade somewhat for models having no exponentially distributed processes but is yet highly reasonable for quick-look analyses. The following table gives the approximate frequency relative errors were below a given error level.

Table 4.27 The approximate frequency relative errors fell below specific error levels.

Error Level	≤10%	≤15%	≤20%	≤25%
Exponential Arrivals and Nonexponential Service	91%	95%	99%	100%
Nonexponential Arrivals and Nonexponential Service	82%	89%	96%	100%
Single-class	84%	98%	98%	100%
Multi-class	90%	94%	98%	99%
Overall	89%	94%	98%	100%

5. Conclusions and Future Research

This thesis has proposed an analytical, stochastic model for the Joint Reception, Staging, Onward Movement, and Integration process, the final phase of U.S. military deployment. It was noted that the process often experiences bottlenecks because pax and cargo entities are not directed at an appropriate rate, thus straining available processing capacity and providing the enemy a vulnerable target. Existing capabilities for providing critical process performance measures, such as throughput and closure, to combatant commanders were explored revealing that many are capable but none can obtain results without lengthy simulation run times.

An analytical, stochastic model of the process was proposed that provides computationally expedient estimates for critical performance measures at points within the process and for the process as a whole. A notional JRSOI model was developed based on information from relevant DoD publications and subject matter experts. Four variations of an open, feed-forward queueing network were presented. A queueing network analysis (QNA) routine was developed for computing the results. Results obtained using QNA were compared with results from computer simulations to demonstrate the accuracy of the approximation algorithms and were found to be extremely accurate in most cases.

The most conspicuous finding of this research was the contrast in run time for QNA as compared to simulation. The run times for discrete-event simulation ranged from 5 to 16 hours, whereas those for QNA were at or below 0.5 seconds. Furthermore, the simulation run times, despite being extensive, were only long enough to ensure that the half-widths for the 95% confidence intervals were less than 5% of the size of the confidence interval means. Much longer run lengths would be required to significantly increase the precision of the simulated benchmarks.

The tests conducted verified that QNA produced exact answers for $M/M/1$ and $M/G/1$ queues. There was a noticeable variance in relative error between QNA and the simulation depending on the queue structure, network dependency, squared coefficients of variation, parametric probability distributions, and traffic intensity particular to each node. In general, QNA performed reasonably well for individual nodes and the entire system for the models tested. Most relative errors were at or below 10% with an upper bound of about 25% relative error. It is concluded that QNA tends to perform better when squared coefficients of variation are close to a value of 1.0. Due to the narrow range of traffic intensities specified, little conclusive information regarding their impact on performance was revealed. It was observed, however, that extremely high traffic intensities led to very large confidence interval half-widths, thus degrading the precision of the associated simulated benchmarks. As discovered by Whitt [29], the accuracy of the approximations does not necessarily decrease as the complexity of the network increases. Some of the most accurate results of this study were obtained when testing the largest and most complex networks. Surprisingly, QNA performed significantly better for the multi-class models than for the single-class models according to the 10% relative error standard (see Table 4.27. However, the performance of QNA for the single-class models was generally superior to that for the multi-class models according to the 15% relative error standard). It is concluded that the distribution chosen to obtain a particular squared coefficient of variation significantly impacts performance. This indicates that QNA could be improved by incorporating higher-order moments into its calculations.

Early tests indicated that Arena's® built-in capability to generate random variates is more effective for some distributions (e.g. gamma) than for others (e.g. beta). In fact, convergence to known steady-state results could not be obtained for a simple $M/G/1$ queue with beta distributed service times until a new beta random variate generator was developed utilizing the built in gamma variate generator. This highlighted the fact that simulations, while providing good general standards against which to compare the QNA results, are not infallible.

This thesis contributes significantly to the military operations research literature by providing a novel approach for modeling JRSOI that is expedient, inexpensive, and produces highly accurate results in contrast to the high cost and extended run times of existing models. It lays the groundwork for the development of a comprehensive queueing network analysis tool specifically tailored for JRSOI. The base model developed here can readily be expanded and adapted to any potential area of conflict. Furthermore, it validates the use of analytical approximation techniques for obtaining performance metrics. The capability to expediently compute network performance measures can greatly increase a combatant commander's ability to properly regulate entity flow and obtain other useful information to aid the efficient build up of combat capability.

There are many potential extensions to this research. First, much additional testing and analysis can be conducted. Simulation run times can be significantly increased, allowing the simulations more time to converge to true steady-state results. This would provide a better standard against which to compare the QNA estimates. Alternatively, multiple simulation runs at the current run length and using distinct random number streams would help reduce statistical variance and thus yield more precise standards. Furthermore, the approximation algorithms can be tested against a wider variety of network topologies, resolutions, arrival and service rates, parametric distribution families, number of classes, number of servers, squared coefficients of variation, and traffic intensities. Extensive testing in the form of designed experiments could reveal patterns indicating if, when, and how these factors (or interactions thereof) affect performance. Such analyses could lead to the placement of restrictions on model inputs or to correction methods for the algorithm. In addition, the effect multiple servers and bulk arrivals and departures have on performance could be investigated.

Second, this research can be extended by exploring the behavior of superposition and departure operations at queueing stations. Superposition occurs when multiple streams of entities arrive at the same queueing station and departure occurs as entities depart a queueing station. Whitt [28] utilized parameters in his algorithms that can be adjusted to account for the impact these operations have on performance. Yet, these parameters were fixed at a default value of zero and the author does not provide a methodology for selecting them. The only guidance provided is that they range from zero to unity.

Third, this research can be extended to explore the possibility of developing an algorithm to obtain performance measures for JRSOI networks based on transient rather than steady-state analysis. If found to be practical, this would provide a more accurate representation of true network activity at a given point in time. However, transient analysis of queueing networks is not straightforward.

Finally, the ultimate practical extension of this research would be to develop a user-friendly software program based on the QNA algorithms for use during deployment. Such a tool would equip analysts in the field with a very inexpensive, portable, accurate, and high-speed analysis capability they could use to assist in the decision making process.

Appendix A. Probability Distributions

Distribution 1 *The exponential distribution, with scale parameter $\lambda \geq 0$, is denoted by expo(λ). Its cdf is given by*

$$F(x) = \begin{cases} 1 - e^{-\lambda x}, & \textit{if } x > 0 \\ \\ 0, & \textit{otherwise.} \end{cases}$$

The mean and variance are given by λ and λ^2 respectively.

Distribution 2 *We denote the gamma distribution with shape parameter $\alpha > 0$ and scale parameter $\beta > 0$ by gamma(α, β). Its cdf is given by*

$$F(x) = \begin{cases} 1 - e^{-x/\beta} \sum_{j=0}^{\alpha-1} \frac{(x/\beta)^j}{j!}, & \textit{if } x > 0 \\ \\ 0, & \textit{otherwise.} \end{cases}$$

The mean and variance are given by $\alpha\beta$ and $\alpha\beta^2$ respectively.

Distribution 3 *The lognormal distribution, with shape parameter $\sigma > 0$ and scale parameter $\mu \in (-\infty, \infty)$, is denoted by LN($\mu, \sigma^2$). There is no general closed form for the cdf so the density function is given as follows*

$$f(x) = \begin{cases} \frac{1}{x\sqrt{2\pi\sigma^2}} \exp \frac{-(\ln x - \mu)^2}{2\sigma^2}, & \textit{if } x > 0 \\ \\ 0, & \textit{otherwise.} \end{cases}$$

The mean and variance are given by $e^{\mu + \sigma^2/2}$ and $e^{2\mu + \sigma^2}(e^{\sigma^2} - 1)$ respectively.

Distribution 4 *The Weibull distribution, with shape parameter* $\alpha \geq 0$ *and scale parameter* $\beta \geq 0$*, is denoted by Weibull(*α, β*). Its cdf is given by*

$$F(x) = \begin{cases} 1 - e^{-(x/\beta)^\alpha}, & \text{if } x > 0 \\ \\ 0, & \text{otherwise.} \end{cases}$$

The mean and variance are given by $\frac{\beta}{\alpha}\Gamma(\frac{1}{\alpha})$ *and* $\frac{\beta^2}{\alpha}\left\{2\Gamma(\frac{2}{\alpha}) - \frac{1}{\alpha}\left[\Gamma(\frac{1}{\alpha})\right]^2\right\}$ *respectively where* $\Gamma(\alpha)$ *is defined by* $\Gamma(u) = \int_0^\infty t^{u-1}e^{-t}dt$ *for any real number* $u > 0$*.*

Bibliography

1. *Arcent Kuwait Mission* (January 2002). http://www-ku.arcent.army.mil/history/index.html.

2. *Arcent Quatar Mission* (July 2002). http://www-qa.arcent.army.mil/news/archive/2002.

3. Bernett, Donald (July 1995). What the Heck is RSOI? Newsletter No. 95-10. Center for Army Lessons Learned, http://call.army.mil/products/newsltrs/95-10/ntcf_toc.htm

4. Bolch, Gunter, Stefan Greiner, Hermann de Meer, and Kishor S.Trivedi (1998). *Queueing Networks and Markov Chains.* John Wiley & Sons, Inc., New York.

5. Bronson, Robert C. (July-August 2000). *Improving the Joint Deployment Process.* http://www.almc.army.mil/alog/issues/JulAug00/MS562.htm

6. Byrnes, Kevin P. and David F. Melcher (March/April 1998). Projecting Combat Power. *Military Review*, **78.2**, 70-72.

7. Cirrincione, Bonnie C. (April 1997). *Joint Total Asset Visibility A Catalyst for Change in Logistics, RWP029.* Air War College, Maxwell Air Force Base AL.

8. Department of the Army (17 March 1999). *Reception, Staging, Onward Movement, and Integration.* Field Manual 100-17.3. HQ USA, Washington.

9. Department of Defense (13 June 2000). *Joint Tactics, Techniques, and Procedures for Joint Reception, Staging, Onward Movement, and Integration.* Joint Publication 4-01.8. GPO, Washington.

10. Department of Defense (9 April 2002). *Joint Tactics, Techniques, and Procedures for Movement Control.* Joint Publication 4-01.3. GPO, Washington.

11. Department of Defense (7 September 1999). *Joint Training, Exercises, and Assesments.* Joint Publication 3-35. GPO, Washington.

12. Department of the Navy (28 February 2001). *Marine Corps Operational Level Logistics.* MCWP 4-12. HQ USN, Washington.

13. Defense Transportation Regulation (DTR) DOD Regulation 4500.9-R-Part III Mobility Nov 2001.

14. Drummer, Donald G (2000). *An Assessment of Modelling and Simulation Tools for Force Projection.* 20000526 074 U.S. Army War College, USAWC Class of 2000.

15. Ennis Jr., Charles W. and Michael T. Mahoney (September/October 1996). Integrating Force Projection in NTC Rotations. *Military Review*, **76.5**, 10-16 .

16. *Fort Riley's Task Force 1-34 prepares to move into Kabal* (August 2001). http://www.kuwait.army.mil/news/archive/2001/article_090701_2.html.

17. *Guide to Automatic Identification Technology (AIT) Use in Force Projection Operations* (August 2000). http://www.deploy.eustis.army.mil/ait/ait/Guide to AIT/AITG-TOC.htm.

18. *Deployment FAQ* (October 2002). http://www.jdtc.transcom.mil/Deployment FAQ/faqpage13/Oct02.

19. Kendall, D (1951). Some Problems in the Theory of Queues. *Journal of the Royal Statistical Society*, **Series B, 13**, 151-185.

20. Kleinrock, Leonard (1975). *Queueing Systems, Volume I: Theory.* John Wiley & Sons, New York.

21. Law, James W (July 1997). *Just-In-Time (JIT) Command and Control (C2) Technical Report/Simulation.* CCDoTT Project, Sponsored by USTRANSCOM. BTG Inc., San Diego.

22. Lidy, Martin A., Douglas P. Baird, John M. Cook, and William J. Sheleski (Dec 1996). *Force Deployment Rock Drill After Action Report, Document D-1954.* Institute for Defense Analysis, Alexandria VA.

23. *Lucky Sentinal* (May 2002). http://www.globalsecurity.org/military/ops/lucky-sentinal.htm.

24. Meyer, Dwain A. (Jan/Feb 2001). Transportation Strategy. *Military Review, Command & General Staff College.* http://www.cgsc.army.mil/milrev/english/JanFeb01/meyer.asp.

25. *S. Korea, U.S. to Hold Joint Military Drills April 21-27* (4 March 2002). Asian Political News. http://www.findarticles.com/cf_0/m0WDQ/2002_March_4/84260214/print.jhtml

26. *Staging/restaging facility has new home in Babenhausen*(Dec 1996). Army News Service http://www.dtic.mil/armylink/news/Dec1996/a19961205ameddawd.html.

27. Schrady, David (Summer 1999). Combatant Logistics Command and Control for the Joint Force Commander. *Navy War College Review*, LII.3.

28. Whitt, Ward (1983). The Queueing Network Analyzer. *The Bell System Technical Journal,* **62.9**, 2779-2815.

29. Whitt, Ward (1983). Performance of the Queueing Network Analyzer. *The Bell System Technical Journal,* **62.9**, 2817-2843.

Vita

Captain Nathan P. Sherman was born in Fort Atkinson, Wisconsin. He graduated as the New York State, Section III Scholar-Athlete of his class from Rome Free Academy High School in Rome, New York in 1991. He attended Mohawk Valley Community College in Utica, N.Y. on a full scholarship and graduated with highest honors earning an Associate of Science Degree in Mathematics in 1993. He graduated Magna Cum Laude from Binghamton University (SUNY) in Binghamton, New York with a Bachelor of Science degree in Mathematics and a Russian minor in 1996.

Captain Sherman received his commission through Officer Training School at Maxwell AFB in Montgomery, Alabama on August 14, 1998. Upon completing Acquisitions Technical School at Lackland AFB in San Antonio, Texas he was stationed at Eglin AFB in Fort Walton Beach, Florida. He was first assigned to the 96th Communications Group where he served as the Drone Control Systems Manager. He was responsible for managing the Gulf Range Drone Control System (GRDCS), the software package used to control Air Force target drones. He was next assigned to the Fuzes Branch, Munitions Directorate of the Air Force Research Laboratory (AFRL) where he served as Research Analyst. He led technical projects for bomb damage assessment and point burst fuzes to include the Air Force Chief of Staff sponsored Targets under Trees (TUT) program. In addition, he managed a program for target detection and classification using active optical imaging.

In 2001, Captain Sherman entered the Air Force Institute of Technology (AFIT) where he worked toward a Master of Science degree in Operations Research, specializing in the area of analytical, stochastic modeling. He was awarded the Dean's Award (and corresponding nomination for the Commandant's Award) recognizing his thesis as the best in the Department of Operational Sciences for his year group. He will remain at AFIT to work towards a Doctor of Philosophy degree in Operations Research.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 074-0188*

1. REPORT DATE *(DD-MM-YYYY)* 01-03-2003	**2. REPORT TYPE** **Master's Thesis**	**3. DATES COVERED** *(From – To)* Jun 2002 – Mar 2003

4. TITLE AND SUBTITLE A STOCHASTIC MODEL FOR JOINT RECEPTION, STAGING, ONWARD MOVEMENT, AND INTEGRATION (JRSOI)	**5a. CONTRACT NUMBER** **5b. GRANT NUMBER** **5c. PROGRAM ELEMENT NUMBER**
6. AUTHOR(S) Sherman, Nathan P., Captain, USAF	**5d. PROJECT NUMBER** **5e. TASK NUMBER** **5f. WORK UNIT NUMBER**

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 P Street, Building 640 WPAFB OH 45433-8865	**8. PERFORMING ORGANIZATION REPORT NUMBER** AFIT/GOR/ENS/03-21
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) USTRANSCOM (Attn: J5AI) Attn: Lt Col Robert T. Brigantic 508 Scott Dr., Bldg 1900 DSN: 779-1459 Scott AFB, IL 62225 e-mail: robert.brigantic@hq.transcom.mil	**10. SPONSOR/MONITOR'S ACRONYM(S)** **11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

A stochastic model for the performance evaluation of a key phase in the deployment process, namely Joint Reception, Staging, Onward Movement, and Integration (JRSOI) is presented. The process is modeled as an open, multi-class tandem queueing network wherein personnel and various classes of cargo are modeled as the flow entities and the stages of the process constitute individual queueing stations. Single- and multiple-class models at both low and high resolutions are presented. No analytical stochastic model of this process currently exists in the literature or in practice. The model provides a quick look at key aggregate performance measures such as system throughput and closure, and can be used to expediently identify problems occurring during JRSOI and the impact they have on the process. This information can substantially aid decision makers in regulating process flow. The queueing network model developed here can easily be expanded and adapted to any potential area of conflict. Numerical comparisons with Monte-Carlo simulation demonstrate that the model provides a viable, novel approach to the problem.

15. SUBJECT TERMS

Stochastic Processes, Networks, Queueing Theory, Mobilization, Military Operations, Joint Military Activities, Queueing Networks, Deployment, Mobility Modeling, JRSOI, RSOI, RSO&I

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Jeffrey P. Kharoufeh, Ph.D. (ENS)
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-6565, ext 4336; e-mail: Jeffrey.Kharoufeh@afit.edu
U	U	U	UU	99	